UNIVERSIDAD CARLOS III DE MADRID

TRABAJO FIN DE GRADO

ESTUDIO DEL RENDIMIENTO BIOMÉTRICO DE DISPOSITIVOS DE HUELLA DACTILAR: ANÁLISIS DE CALIDAD Y VARIABILIDAD DE LA MUESTRA

GRADO EN ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

AUTOR: PABLO FERNÁNDEZ LÓPEZ

Tutora: María Belén Fernández Saavedra

RESUMEN

El trabajo presentado se centra en el estudio de rendimiento de sensores biométricos basados en huella dactilar. El estudio concretamente analiza la influencia de la calidad de la huella. Para poder determinar la calidad de estas huellas se ha usado el algoritmo NFIQ, este algoritmo asigna a cada muestra biométrica un valor de calidad. Estos valores se han distribuido entre tres niveles de calidad (alto, medio y bajo), dividiendo de esta forma las muestras. Finalmente, se han desarrollado tres estudios de rendimiento, uno por cada nivel de calidad, y se han comparado los resultados.

Para poder ver que la influencia de NFIQ es equivalente en cualquier sensor biométrico, el estudio ha sido llevado con tres sensores de tres fabricantes distintos. Además, para poder tener suficientes valores se ha pedido la colaboración de 50 usuarios, obteniendo más de 6000 muestras.

Una vez obtenidas las muestras, éstas han sido procesadas haciendo comparaciones entre ellas. Estas comparaciones permiten conocer el rendimiento de los sensores para los diferentes niveles de calidad en cada sensor. Estas comparaciones se dividen en dos grupos ya sean comparaciones entre imágenes de un mismo usuario o entre usuarios distintos. Las llamadas comparaciones genuinas y de impostores, respectivamente.

Al obtener estas listas se ha podido representar una serie de valores que indican el porcentaje de usuarios que conseguirían vulnerar el sistema, el valor mínimo de comparación para hacer el sistema más seguro a cambio de ser más restrictivo, etc.

Finalmente, al analizar estas gráficas se observó cómo los valores NFIQ de menor calidad tenían una clara influencia negativa sobre el rendimiento de los sensores; pero los valores de calidad media y alta mostraban resultados muy similares. Es por esto que se plantea nuevos estudios que permitan un mayor análisis del algoritmo NFIQ para discernir si es necesario la mejora del algoritmo para notar mayor diferencia entre las calidades.

EXECUTIVE SUMMARY

The former study is intended to bring a new point of view on biometric identification. It focuses on studying how valid the approximation value of the algorithm NFIQ, made by the NIST, has on the actual overall quality of a biometric system.

The project will use 3 different fingerprint sensors, these sensors are from a different manufacturer each. The purpose of these sensors is to see if the use of different technology has any influence on the NFIQ algorithm, moreover thanks to the sensors the study can assure that the results obtained would not only apply to a specific product but for a great variety ones.

To carry out the research 50 users have been ask to assist by interacting with the three sensors. Each user generated at least 3 Enrol patrons with each finger, from thumb to the middle finger, in addition the users will go through the process of verification 6 times from each fingers. Both processes was carried out for each sensor.

Once all the images were taken, these would be processes with a C# program. The process consisted on creating a list of genuine comparison values and impostor comparison values for each sensor. To obtain the first list, each finger would be assign a enrol patron, the one with the lowest NFIQ value (therefore the highest quality), and compare that patron with all the verifications images. The set of numbers of all the genuine comparison of each finger would become the final list of genuine comparison of the sensor those images belonged to. To obtain the second list, each finger Enrol would be compared to all the verification images of all other users. This las list was significantly bigger since there were more comparisons to be made. Once again, the set of values would be grouped together and presented as the impostor comparison values of the specific sensor. This process would be carry out for each quality level.

After recollected the data that has been created, the different graphs needed to see the results that had been obtained were represented. When searching for the performance of a biometric sensor it is important to check 3 different graphs: the DET, which represents the

FRR in dependence of the FAR (in percentage); the ROC, which shows for a percentage of admitted user how many would be impostor; and finally the FARvsFRR, which is informs of how the FRR and FAR vary when increasing a threshold value of comparison. For each sensor there would be 1 of these graphs with the values of low, medium and high quality on the same graph. Finally to be able to compare the sensors there would be an extra three groups of graphs comparing the different sensors at different quality levels.

Thanks to this graphs, some conclusion were able to be obtained. As expected the NFIQs 5-4 were very much divided with other qualities. The values obtained at this level so very poor quality to the point of being of no use in a real life system. Nevertheless, the NFIQs of value 3 were very much a surprise due to the great results obtained. These values were comparable to those of value 1-2 even to the point of matching them in some cases.

On the final conclusions, it was reached that the NFIQ algorithm still has place for improvement. The difference between the low quality values and the medium quality were too high in comparison with the difference with medium and high. It could be of interest to try to improve the NFIQ algorithm so that these values are better distributed.

As a conclusion dedicated for the future companies that want to include biometric recognition as part of their security system, it was suggested to encourage a study on what are the exact influences on the NFIQ values. This means, what environmental or biological characteristics makes one fingerprint better than the other. If the factors that lower the quality of the images can be taken out, there will be more warranties of using these sensor in a security system.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1. INTRODUCCIÓN

El presente trabajo pretende tomar una nueva visión en la forma de realizar estudios de rendimiento en sensores biométricos.

Los sensores biométricos están empezando a ser cada vez más relevantes en el mundo de los controles de acceso, lo que ha hecho que su rendimiento haya ido adquiriendo una mayor estandarización en su visión del rendimiento.

Estos rendimientos se centran fundamentalmente en reducir la probabilidad de confundir una persona con otra, rechazar a una persona que intenta acceder al sistema de manera irregular o no autorizada y gestionar el porcentaje de que estos dos casos ocurran dadas unas determinadas limitaciones.

Este trabajo pretende contribuir a comprobar si hay cabida en la mejora de los estudios del rendimiento o uso de estos sistemas.

Estos estudios se centran, en gran medida, en las ocasiones en las que la característica biométrica tiene un mínimo de calidad concreto.

Este trabajo pretende comprobar cómo la calidad de estas características influye realmente en el rendimiento de los sistemas.

Para ello, se han realizado los análisis que normalmente pasarían estos sensores, pero dividiendo los datos según la calidad de los mismos.

El trabajo se centra en los sistemas biométricos basados en las huellas dactilares. Como su nombre indica, los sistemas de huella dactilar almacenan imágenes de las huellas dactilares de sus usuarios para, más adelante, poder hacer comparativas entre imágenes y así tener poder de decisión sobre quién accede al sistema.

El trabajo se puede dividir en tres partes o procesos:

- *Recolección de datos*. En este proceso, se ha conseguido la ayuda de cerca de 600 voluntarios para que hagan uso de los sensores biométricos y así poder tener la cantidad de imágenes necesaria.

- *Tratamiento de datos*. Una vez obtenidas las imágenes, se ha desarrollado un programa en C# capaz de hacer todas las comparativas necesarias para tener unos datos que reflejen los distintos rendimientos de los sensores. Se ha escogido el lenguaje C# debido a que las funciones a utilizar están pensadas para ser compiladas con un programa de este lenguaje.

- *Análisis de datos*. Una vez obtenidos los datos finales, se podrá hacer un análisis final de los rendimientos de los sensores a través del programa Matlab. Este programa utiliza un lenguaje de alto nivel en el que se permite visualizar datos numéricos así como escribir algoritmos y procesar variables numéricas. Como añadido existen funciones escritas en Matlab que permiten representar las gráficas necesarias para el estudio con el uso de una simple función.

1.1 Motivación

La motivación para realizar este estudio se debe a la gran relevancia que la identificación biométrica está tomando día tras día, sustituyendo sistemas de seguridad convencionales por sistemas en los que se puede reconocer a un individuo sin necesidad de códigos ni tarjetas.

Como cualquier nuevo elemento tecnológico es necesario que se hagan estudios para verificar la viabilidad de estos productos para mejorarlos. En los sistemas biométricos se ha usado el algoritmo NFIQ para evaluar la calidad de las muestras tomadas, dándoles un valor numérico para indicar si la muestra puede ser usada o no.

El motivo de este estudio es comprobar cómo de fiable es el algoritmo NFIQ y si se podría mejorar para que dé una mayor información en su categorización de las muestras.

Dependiendo de los resultados del proyecto se podría plantear la mejora del algoritmo o indicar cuando el valor NFIQ de una muestra puede influir realmente en el sistema.

1.2 Objetivos

En línea con lo expuesto en el apartado anterior, este trabajo tiene como objetivo principal la realización de un análisis del rendimiento de sensores biométricos según la calidad de la muestra.

Este tipo de trabajos se suele llevar a cabo a través de grupos de investigación, como el Grupo Universitario de Tecnologías de Identificación (GUTI) con el fin de asegurar la realización de estudios imparciales.

Concretamente, el presente trabajo ha sido realizado en colaboración con el GUTI.

De forma adicional al análisis general de rendimiento de sensor, se pretende realizar este análisis en grupos de imágenes categorizadas por la calidad de las mismas. De esta forma, todos los sensores tendrán tres estudios de rendimiento: para imágenes de baja, media y alta calidad.

Como añadido, los datos obtenidos durante el proyecto serán usados para futuros estudios del GUTI. Estos estudios se componen en su mayoría en estudios estándar de rendimiento de sensores.

Finalmente, se hará una comparativa entre los sensores utilizados y se comprobará si todos responden de la misma forma a los distintos niveles de calidad utilizados. Con estos resultados se podrá saber si las diferentes tecnologías de los sensores responden de forma distinta a las calidades.

El estudio, por lo tanto, pretende esclarecer la utilidad de los sensores biométricos en el mundo de los sistemas de seguridad y comprobar si tienen cabida en él. De forma adicional, se comprobará si el sistema de diferencia de calidad en las muestras podría ayudar a mejorar estos sistemas de forma significativa.

1.3 Marco Socio-Económico

La importancia de la identificación biométrica durante estos años se ha vuelto casi exclusiva de los sistemas de seguridad.

En los últimos años se ha visto que los sistemas de seguridad convencionales llevados por clave de seguridad suelen ser fácilmente vulnerados. Esto se debe a que un código, a pesar de ser único, puede ser descubierto y distribuido de varias formas. Este problema se ha intentado solucionar añadiendo complejidad a los códigos con las claves WPA y WPA2, pero debido a la complejidad de estas claves puede ser pesado el memorizarlas o escribirlas.

En este mercado de seguridad también han aparecido otras formas de seguridad como pueden ser las tarjetas *Radio Frequency IDentification* (RFID) las cuales no requieren uso de memoria por parte del usuario. Este sistema de seguridad a probado ser de gran utilidad pero sigue teniendo problemas de tarjetas extraviadas o robadas.

Como solución alternativa se está llevando cada vez más al mundo de la seguridad los sistemas biométricos. Un sistema biométrico no requiere códigos ni tarjetas, tan sólo requiere tener una base de datos para poder reconocer a sus usuarios. Un usuario es reconocido por sus características fisiológicas lo cual permite, idílicamente, que ese usuario pueda ser distinguido de cualquier otra persona.

De poder realizar con completa seguridad un sistema biométrico, se podría crear un sistema seguro y que requiera muy poco esfuerzo por parte de los usuarios. Esto permitiría un descenso en vulnerabilidades de los sistemas y un paso más fluido de los usuarios con premiso de acceso.

Este estudio se hace en pos de ayudar a mejorar los sistemas de seguridad biométricos. Se espera que con los datos reunidos se puedan mejorar estos sistemas para asegurar una mayor seguridad y una mayor fluidez en su paso.

1.4 Marco Regulatorio

Para la realización de un estudio como el realizado hay que tener en cuenta que se trabaja con datos personales de una serie de voluntarios. Por lo tanto todos los datos tienen una regulación de privacidad para defensa de los usuarios. Además de esta protección de datos también hay que tener en cuenta que un estudio de rendimiento se tiene que ajustar a una normalización para asegurar que los datos del estudio se puedan comparar con otros de forma directa.

Según la directiva relativa a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos se considera datos personales como: "toda información sobre una persona física identificada o identificable (el "interesado") ; se considerará identificable toda persona cuya identidad pueda determinarse, directa o indirectamente, en particular mediante un número de identificación o uno o varios elementos específicos, característicos de su identidad física, fisiológica, psíquica, económica, cultural o social".

Por lo tanto todos los datos biométricos quedan bajo la norma de protección de datos personales y han de ser tratados como tales. El tratamiento de estos datos y su distribución queda por lo tanto bajo la Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal y Real Decreto 1720/2007, de 21 de diciembre, por el que se aprueba el Reglamento de Desarrollo de la LOPD. En esta norma se establece que:

- Se debe de inscribir los ficheros en el Registro General de la Agencia Española de Protección de Datos (AEPD).
- Se debe de informar a los afectados.
- Se debe de obtener el consentimiento de los afectados.
- Se debe ejercer el derecho de los afectados

En lo que refiere a los pasos a seguir durante el estudio se ha seguido la normativo ISO 19795. Esta norma de estandarización define como se debe llevar el proceso de recolección de datos y cómo se han de representar los valores para su posterior análisis. Esta norma

permite que todos los estudios representen sus resultados en un mismo marco lo cual hace que las conclusiones sean más objetivas.

1.5 Estructura

El trabajo se divide en una serie de capítulos, se comienza por explicar en qué consiste la biometría para aquellos lectores con poco o ningún conocimiento sobre este campo; después se explicará que frente pretende abarcar el estudio; a continuación se verán los pasos llevados para el proceso del trabajo, y finalmente se verán los resultados y se llegarán a las conclusiones. La división de capítulos es la que sigue:

- En el capítulo 2 se explicará en qué consiste un sistema biométrico y los tipos de análisis que se suelen realizar, esto permitirá entender todo el trabajo realizado durante este estudio.
- El capítulo 3 describe en mayor detalle la diferencia entre el estudio realizado y los estudios más tradicionales. También presenta los sensores utilizados durante el trabajo.
- Los capítulos 4 y 5 presentan los pasos realizados durante el proyecto. El capítulo 4 se centra en cómo se tomaron los datos utilizados y el capítulo 5 en cómo se han ido transformando en datos que se pueden analizar.
- En el capítulo 6 se verán una serie de gráficas que representan el resultado obtenido en este estudio. En este capítulo se observarán los datos obtenidos y se harán explicaciones cuantitativas y cualitativas de los mismos.
- Finalmente en el capítulo 7 se sopesará sobre los datos y el significado de los mismos. Asimismo y se enumerarán distintos estudios que se podrían realizar en el futuro en vista del trabajo realizado y los resultados obtenidos.

2. ESTADO DEL ARTE

Para entender el estudio es necesario aclarar una serie de términos y los estudios que han llevado a poder realizar este proyecto. Para ello se hablará en qué consiste la biometría, sus sistemas y como se deben de analizar los mismos.

2.1 Biometría

La Biometría se refiere al campo de la ciencia que estudia estadísticamente parámetros biológicos, lo que abarca desde estudios de ecosistemas, hasta las influencias meteorológicas que pueden producirse en la propagación de una enfermedad, por ejemplo.

La identificación biométrica, o el reconocimiento biométrico, es un campo más concreto de la biometría en el que se pretende identificar a un individuo a través de sus características biológicas, rasgos físicos o comportamientos, de forma más o menos automática.

Un sistema biométrico es aquel en el cual se comprueba la identidad de una persona no por un código o tarjeta, sino analizando de forma automática sus características biológicas.

Estas características pueden referirse a cualidades físicas, como la forma de los ojos o la huella dactilar, o incluso a cualidades de comportamiento, como sería el estudio de la firma o hasta la forma de moverse.

Este trabajo se centra exclusivamente en los sistemas que utilizan identificación biométrica por huella dactilar.

2.2 Sensores biométricos

Se define un sensor biométrico como aquel sensor capaz de extraer datos numéricos para poder diferenciar dos seres biológicos de forma automática.

Cada sensor biométrico utiliza su método propio de toma de imagen. Existen sensores ópticos, térmicos, capacitivos e, incluso, ultrasónicos.

Los sensores utilizados en este trabajo usan el método térmico o capacitivo. A continuación se explican estos dos tipos de tecnologías.

Los sensores capacitivos tienen una superficie conductiva en la cual se almacena una carga eléctrica. En el momento de colocar un dedo sobre ella, el valor capacitivo de la placa cambia, debido a la diferencia de potencial.

Una huella dactilar consiste en valles y cimas que se extienden por el dedo dentro de la propia piel. Estos valles y cimas dan lugar a que exista una diferencia de potencial entre las zonas de unos y otras. Con esta información, el sensor es capaz de hacer un mapa de la forma de la huella dactilar.

Los sensores térmicos se basan en la diferencia de temperatura. La placa de estos sensores es termosensible. Debido a los valles, hay diferencia de espacio entre la placa y la piel, lo cual hace que se produzca una diferencia de temperatura entre las distintas líneas de la huella. Una vez más, con esta información el sensor es capaz de hacer un mapa de ella.

Independientemente del método usado por los sensores, todos buscan las mismas características únicas de cada huella.

La probabilidad de que dos huellas resulten idénticas es de 1 entre 64 billones, cifra que supera con creces la población mundial. La diferencia entre huellas reside en una serie de características únicas de cada huella [1]. Estas características son:

- *Valles y cimas.* Con una inspección simple de su huella, cada persona es capaz de ver una serie de valles y cimas en ella. Los patrones de estos valles no son únicos: se pueden clasificar en distintos patrones, como arco, espiral (tanto en sentido izquierdo como derecho), círculos concéntricos o acabados en tienda de campaña. Al no ser una característica única, muchos sensores no tienen en cuenta estos patrones.

- *Poros.* Con sensores de muy alta precisión, se pueden distinguir patrones en los poros de sudor situados en la punta del dedo. Estos patrones son fáciles de distinguir entre individuos, lo cual los convierte en ideales para poder hacer comparativas. Sin embargo, requieren sensores de muy alta precisión.

- *Minucias*. Las minucias de una huella dactilar son los puntos en los cuales una cima se diverge en dos o termina (Figura 1). Estas minucias no son un mero punto, sino que además tienen una orientación según el camino que debiera seguir la cima. Por lo tanto, cada huella dactilar tiene una serie de minucia que le hacen única. El estudio de minucias es el más común, puesto que no requiere de un sensor de alta calidad.

- *Otras características.* De manera adicional, se podrían tener en cuenta factores como manchas, cortes o cicatrices. Sin embargo, estas características son temporales, lo cual las hace poco fiables.

Figura 1: Minucias en una huella dactilar **[2]**

Debido a que los sensores de huella dactilar pretenden obtener una imagen 2D a partir de un objeto 3D en, pueden ocurrir varios errores (Figura 2).

Por ejemplo, puede generarse "ruido" debido al exceso humedad, la cual inunda la imagen y difumina las líneas de las huellas; a la falta de humedad, sobre todo en relación con los sensores capacitivos, debido a que se elimina el factor del dieléctrico de la humedad de la piel; a la falta o exceso de presión, ya que la presión ajusta el contraste de la imagen; y,

finalmente, a la mala colocación, debido a que las minucias son vectores si se intenta comparar la misma huella, pero con dos ángulos distintos se puede perturbar la comparación.

Figura 2: De izquierda a derecha, huella con poca presión o fría, huella húmeda o con demasiada presión, y huella movida [2]

Para poder regular la calidad de una imagen, el National Institute of Standars and Technology (NIST) decidió que era necesario implantar una escala en función de calidad de una imagen. Este sistema de categorización se denomina NIST Fingnrprint Image Quality (NFIQ). Esta institución se ha dedicado al apoyo de las nuevas tecnologías, ya sea con nuevos descubrimientos, ayuda a la estandarización de productos electrónicos o con estadísticas realizadas sobre estos productos [3]. El NFIQ de una huella es un valor comprendido entre 1 y 5 que indica cuál es la probabilidad de detectar un error debido a la calidad de las minucias [4].

El rango 1 de NFIQ indica una buena calidad de imagen y prevé una menor probabilidad de error. El rango 5, en cambio, indica una baja calidad de imagen y prevé una probabilidad muy alta de error a la hora de hacer las comparaciones entre minucias.

2.3 Sistema biométrico

Una vez expuesto qué datos se obtienen con un sistema biométrico, ahora hay que saber qué se hace con esos datos.

Hay tres posibles finalidades para las cuales se toma un dato biométrico (Figura 3):

- *Reclutamiento*. Para almacenar un usuario en el sistema.
- *Verificación*. Para comprobar si el usuario es una persona en concreto
- *Identificación*. Para saber si la persona está registrada en la base de datos actual

Un sistema biométrico ha de ser capaz de realizar cualquiera de estos procesos.

El proceso de Reclutamiento es el proceso de la toma del patrón biométrico. Este patrón es crucial, puesto que va a ser el valor comparativo para lo toma de decisiones en los procesos de Verificación e Identificación.

En el proceso de Reclutamiento se toma una imagen de huella, se comprueba la calidad de la misma a través de su NFIQ, se genera un patrón biométrico y se almacena todo ello en una base de datos.

Normalmente, este proceso se realiza más de una vez para poder tener más de un patrón con el que trabajar y poder así seleccionar el de mayor calidad.

El proceso de Verificación se puede definir como el proceso de comparación 1:1.

En este proceso, el sistema toma dos datos: el patrón a usar y la imagen de un sensor biométrico. El sistema escoge el patrón de un usuario del cual se va a realizar la comparación. A continuación, coge una nueva imagen a través del sensor y hace la comparativa. De esta forma, el sistema compara ambas imágenes y da un valor de aceptación.

Dependiendo de un valor umbral, o valor mínimo de aceptación, el sistema decide entonces si considera que el usuario y el patrón se corresponden. Este valor umbral dependerá de una serie de valores estadísticos que se explicarán más adelante.

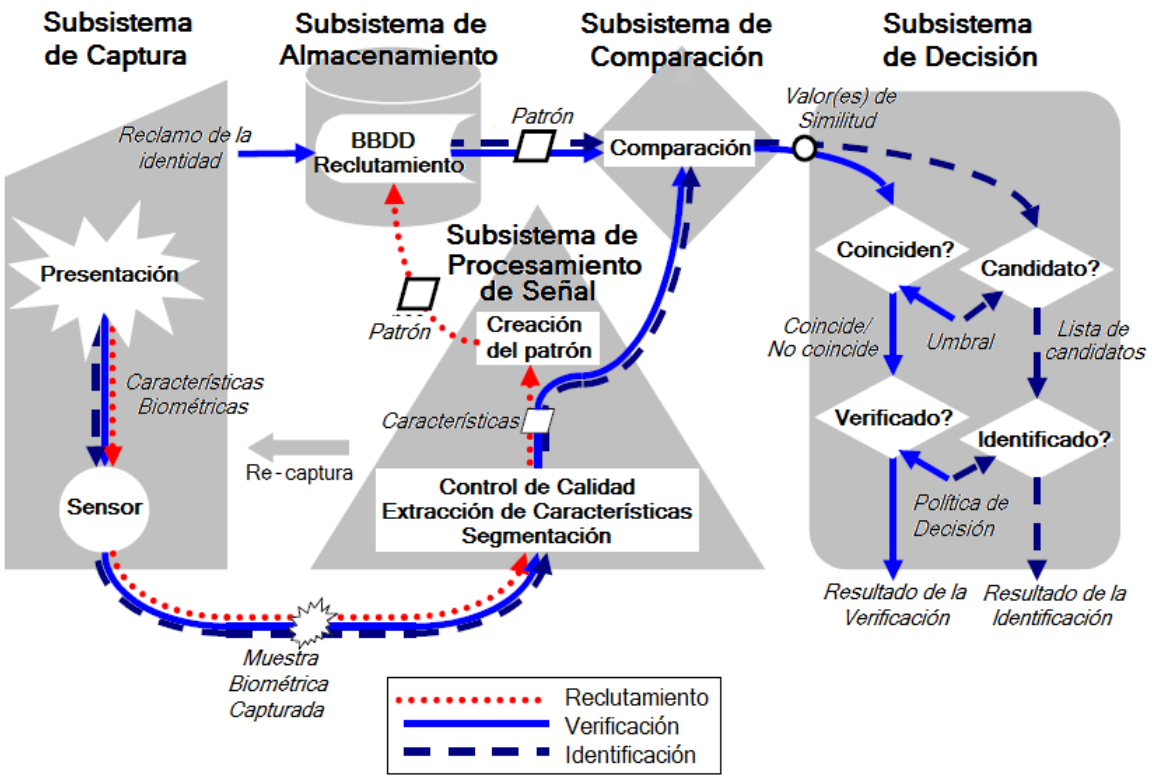


Figura 3: Procesos de un sistema biométrico [2]

El proceso de Identificación es similar al de Verificación pero en valor de comparativa 1:N.

En este proceso, el sistema toma una imagen del usuario a través de un sensor. Después, yendo uno por uno, compara con todos los patrones almacenados en la base de datos y crea una lista de candidatos, de tal forma que, cuando finaliza, el sistema tiene una lista de candidatos con un valor de aceptación. Escogiendo el valor más alto, se decide a qué usuario se corresponde.

De forma adicional, se le puede asignar un valor umbral de aceptación. Al añadir este umbral, se puede dar el caso de que el usuario sea rechazado al no pertenecer a la base de datos.

2.4 Rendimiento de sensores biométricos

En la Biometría, se pueden realizar distintos estudios, tales como rendimiento, seguridad, usabilidad, etc.

El presente trabajo se ha centrado exclusivamente en el estudio de rendimiento y se ha realizado sobre tres sensores de huella dactilar.

El rendimiento de un sensor biométrico se centra en dos cualidades principales: la probabilidad de rechazar un usuario "genuino" y la probabilidad de aceptar un usuario "impostor". Se dice que un usuario genuino cuando se comparan las minucias del usuario con las minucias de su propio patrón. Usuario impostor se define como aquel que compara sus minucias con el patrón de otro usuario. Por lo tanto el rendimiento de un sensor se centra en estudiar como el valor umbral influye en el número de usuarios genuinos rechazados y el número de usuarios impostores aceptados. Este rendimiento se puede estudiar de distintas formas: porcentaje de usuarios impostores en un lote de usuarios aceptados, influencia del valor umbral en el número de usuarios genuinos aceptados y usuarios impostores rechazados, porcentaje de usuarios impostores rechazados a expensas de valores genuinos rechazados, etc.

Hay distintos tipos de tasas que influyen en el rendimiento de un sensor biométrico. Algunas de ellas son:

- *Tasa de fallo de reclutamiento* (FTE). Esta tasa informa del número de usuarios que no consiguen dar un patrón aceptable.

  Como se explica en el apartado de proceso, cada usuario tiene que dar dos imágenes válidas para generar un patrón en la base de datos. Para ello, dispone de tres intentos por imagen. Si, a pesar de ello, no se consigue generar un patrón viable para el usuario, la tasa FTE aumenta.

- *Tasa de fallo de adquisición* (FTA). Indica el número de intentos que ha tenido que hacer un usuario hasta que el sistema ha conseguido obtener una huella con la suficiente calidad.

  Para obtener los valores de la FTA, el propio sistema almacena el número de intentos que ha tenido que hacer el usuario para aceptar la imagen. Este dato se corresponde no sólo al número de intentos por usuario, sino al número de intentos por cada uno de sus dedos.

  Este error se basa únicamente en que la calidad de la imagen sea suficientemente baja en la escala de NFIQ. Otros errores, como el de falsa aceptación y falso rechazo los cuales se explican más adelante, no se tienen en cuenta.

- *Tasa de falso rechazo* (FRR). Señala la proporción de intentos de reconocimiento genuinos para los cuáles el sujeto es rechazado.

  Uno de los valores que más interesa en un sistema de seguridad cualquiera es que no se entorpezca el acceso de las personas que tienen permiso para ello. Es por esta razón por la que se tiene que obtener un valor estadístico que cuantifique cuántas veces tiene que repetir el proceso de verificación cada usuario autorizado antes de que se le de paso.

  Esta tasa engloba la tasa FTA y añade a sus factores las imágenes con un valor NFIQ válido, pero que no ha coincidido con el patrón del usuario.

- *Tasa de falsa aceptación* (FAR). Indica el número de intentos de impostores que son aceptados por el sistema.

  Por supuesto, todo sistema de seguridad debe impedir el acceso de impostores, ya que esta es una de sus funciones más relevantes. Por ello, se ha de realizar un estudio que compare un patrón con todos los usuarios del sistema, menos el genuino, para comprobar el porcentaje de usuarios que serían capaces de vulnerar al sistema.

Estas tasas son fuertemente dependientes del valor umbral escogido. Para poder hacer un estudio exhaustivo de la influencia que tiene, existe una normativa ISO [5] que presenta una serie de gráficas que hacen que el estudio de rendimiento resulte más sencillo. La *gráfica de densidad de probabilidad* es la que habitualmente ayuda a encontrar el valor umbral de los sistemas. En ella, están representados en el eje de abscisas los valores de comparación obtenidos; y en el eje de coordenadas, la cantidad de veces que se ha obtenido ese valor.

En esta gráfica se representan dos funciones:

- La *tasa de aceptación verdadera* (TAR). En ella se representan las comparativas genuinas (usuario real con el dedo correcto).

- La *tasa de rechazo verdadero* (TRR). En esta se representan las comparativas impostoras (usuario impostor con cualquiera de sus dedos).

A partir de aquí, se puede tomar el valor umbral de tres formas distintas, según el nivel de seguridad que se quiera tener:

- *Alta seguridad*. Se toma el valor umbral a partir del cual no existen impostores.

  Gráficamente, se vería desplazando el valor hacia la derecha. Esto crea sistemas en los que el acceso de impostores se acerca en gran medida a 0; sin embargo, también significa que una proporción de usuarios genuinos queda rechazada.

- *Seguridad media*. El valor umbral se toma en el punto donde el área de FRR y FAR son iguales (Figura 4).

  De esta forma, se obtiene un equilibrio en el que es posible que haya impostores que accedan al sistema, pero al mismo tiempo se evita rechazar a un número elevado de usuarios genuinos.

  Esta sería la medida más utilizada para sistemas de control de acceso con grandes bases de datos y que protegen materiales de poco valor.

- *Seguridad baja*. El valor umbral se sitúa en el punto donde nace los valores genuinos.

  Gráficamente, se vería desplazado hacia la izquierda. Esto crea sistemas que rara vez rechazan usuarios genuinos, permitiendo así una elevada fluidez de paso. La contrapartida es que el sistema permite el acceso de un gran número de impostores.

  Sistemas con este nivel bajo de seguridad tienen que estar reforzados con un segundo sistema o ser de simple control de acceso para aplicar a continuación un control de paso.

Esta gráfica también se puede usar para comparar sensores, distintas restricciones etc. Al hacer una comparativa entre gráficas, la más deseada será aquella que tenga una mayor separación entre las funciones, siendo la ideal aquella en la cual sus curvas no llegan a cortar.

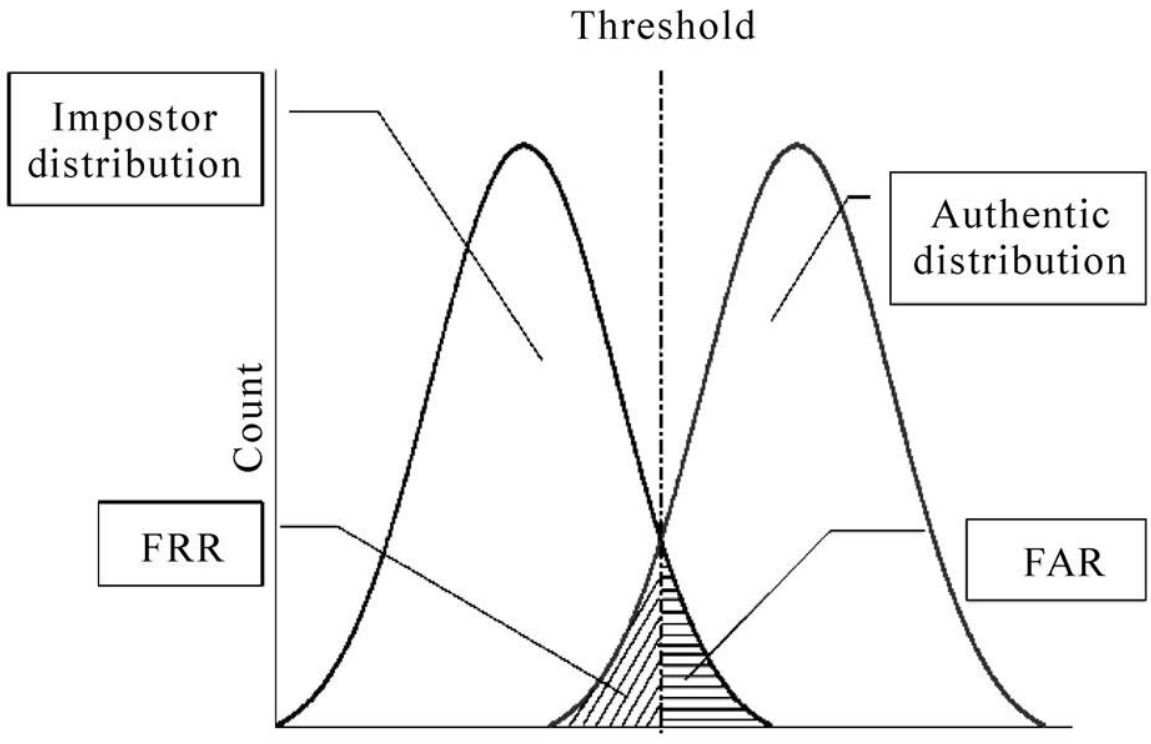


Figura 4: Gráfica de densidad de probabilidad [6]

La curva ROC (Receveing Operating Characteristics) es una gráfica que ayuda ver la proporción de usuarios impostores que son admitidos en un sistema (Figura 5).

En el eje de abscisas se representa la FAR; y en el eje de ordenadas la TAR, es decir la tasa de aceptación verdadera que se calcula como el inverso de la FRR (1-FRR).

En esta curva se suelen representar varios sistemas para hacer comparativa entre ellos. En ella se puede ver, si se quiere limitar el número de usuarios no deseados cómo influye en el rechazo de usuarios genuinos. De tal forma que cuando se selecciona un porcentaje de usuarios no deseados admisible, se puede ver cuántos usuarios genuinos serán rechazados en compensación.

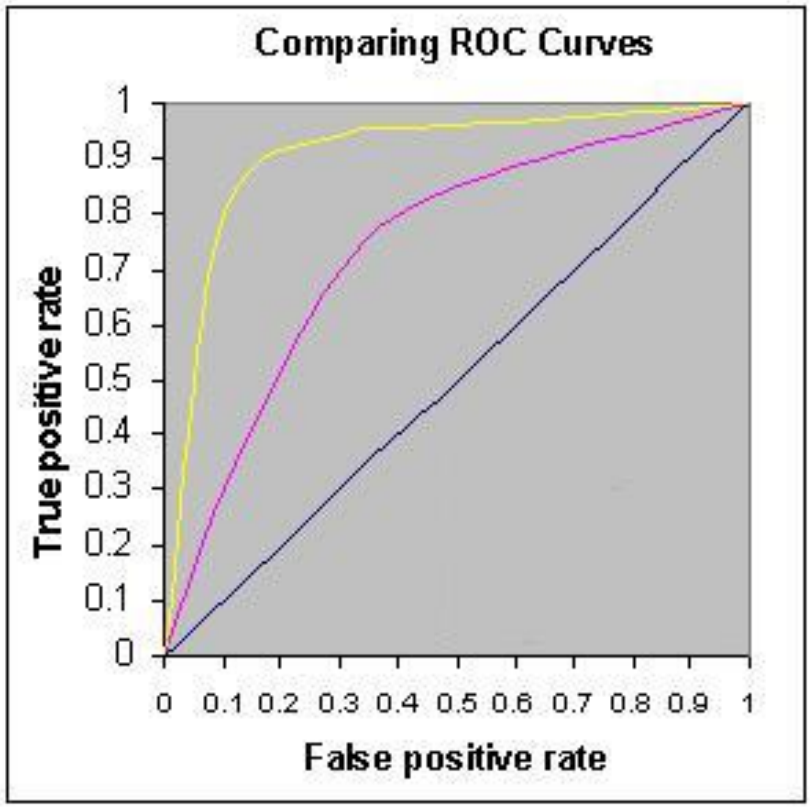


Figura 5:Gráfica ROC (Receveing Operating Characteristics) [7]

Finalmente, la curva DET (Detection Error Trade-off) muestra cómo una selección de umbral afecta a los valores de FRR y FAR (Figura 6). En el eje de abscisas, se sitúa el FAR; y en el eje de ordenadas, el FRR.

Esta gráfica permite una vez más comparar distintos sistemas. En ella se observa cómo cuanto mayor se permita que sea el FAR, menor será el FRR, llegando a una situación en la que se podrá escoger qué valor umbral se debería asumir para conseguir la proporción deseada de impostores admitidos, a cambio de no rechazar un determinado número de usuarios genuinos.

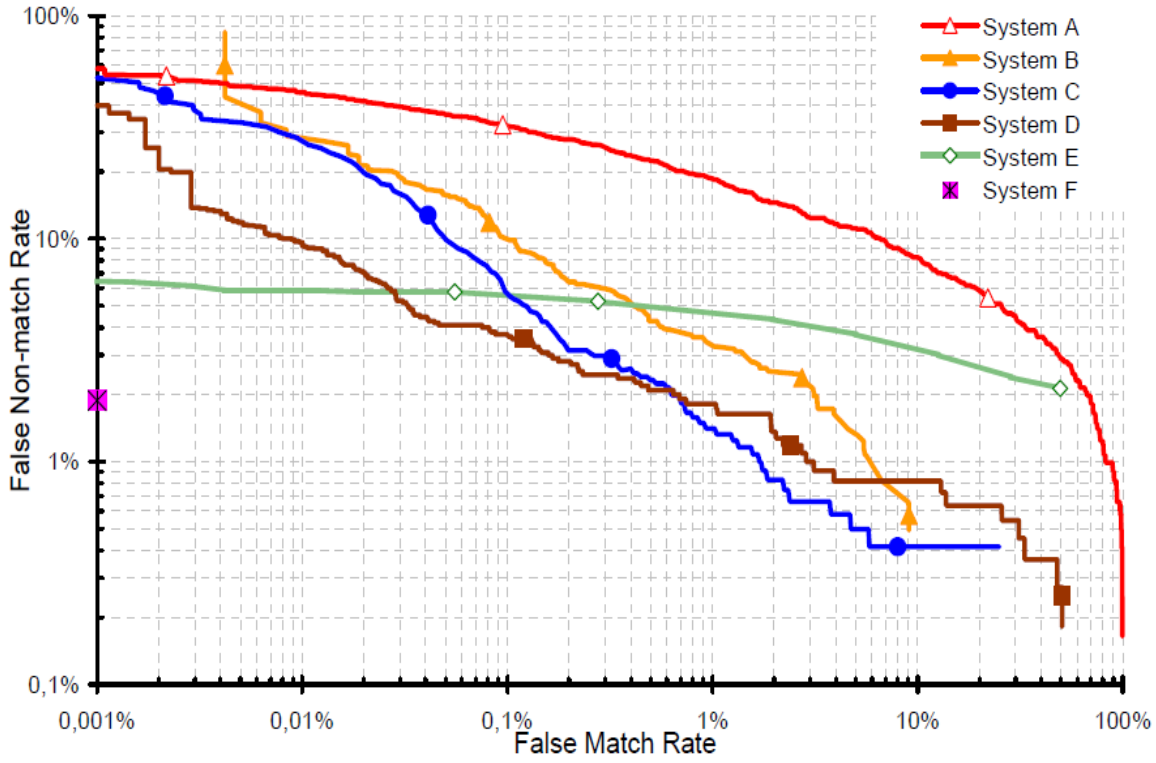


Figura 6: Gráfica DET (Detection Error Trade-off) [8]

3. ESTUDIO EFECTUADO: COMPONENTES DEL SISTEMA

Para la realización completa y detallada de un análisis como el planteado en el presente trabajo, se necesita una amplia base de datos de huellas dactilares. Esta base de datos se suele componer de cientos de personas.

No obstante, para la elaboración de este trabajo se ha contado con la aportación de datos procedentes de 50 usuarios porque una base de cientos de personas exigiría programas de ejecución extremadamente larga y, sobre todo, porque su objetivo fundamental es realizar una primera aproximación al estudio de calidades de huella. Este estudio se podría realizar con más datos para confirmar las conclusiones realizadas.

Según se ha mencionado en apartados anteriores, el trabajo se centra en el estudio de rendimiento de los sensores, según la calidad de los mismos. Por lo tanto, se pretende en él obtener las gráficas de densidad, ROC y DET de cada sensor por cada grupo de calidad.

Para ello, se ha desarrollado una app de Visual Studio en la cual se toman las muestras de los 50 usuarios y se crean las listas de comparación tanto genuinas como de impostores. Una vez generadas estas listas se transforman los datos en las gráficas de densidad ROC y DET utilizando el programa MatLab.

3.1 Sensores utilizados

El estudio se ha realizado con tres sensores distintos:

- UPEK EikonTouch 510 fingerprint sensor (UPK) (Figura 7 a). Este sensor se basa en la tecnología de diferencia de potencial para extraer las imágenes.

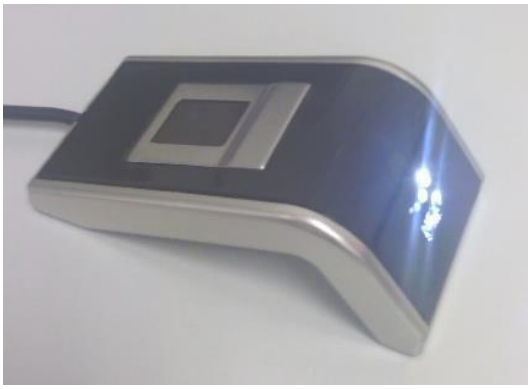
  En otras palabras, se comporta como un condensador, al colocar el dedo sobre el sensor, se registran los diferentes niveles de electrones que hay en la superficie del dedo y se crea una imagen con esta información.

- FPC1011F3 fingerprint sensor (FPC) (Figura 7 b). Al igual que el UPK, este sensor se comporta como un condensador y obtiene las imágenes con las diferencias de potencial en la superficie del dedo.

- NB-3010-U Fingerprint sensor (NXT) (Figura 7 c). Al contrario que los otros dos sensores, este se basa en las diferencias de temperatura de la superficie del dedo y no en sus cargas eléctricas.

(a)

(b)



(c)

Figura 7: Sensores UPK (a), FPC (b) y NXT (c).

3.2 Calidades

Además, el estudio se ha realizado para tres calidades distintas de NFIQ:

- La *calidad alta* corresponde a los NFIQ 1 y 2.

  Estos valores son difíciles de obtener, pero se presupone que, si tan sólo se hace un estudio con estos valores, el rendimiento de los sensores debe ser alto.

- La *calidad media* corresponde a los NFIQ 3.

  Este es el valor aceptable que más comúnmente se obtiene. Los valores de este rendimiento deben ser los que más se acerquen a los estudios habituales de rendimiento de sensores de huella dactilar. F

- La *calidad baja* corresponde a los NFIQ 4 y 5.

  Se espera que los rendimientos de los sensores decaigan drásticamente, puesto que las imágenes de estas calidades suelen ser rechazadas de forma inmediata y no se suelen usar en los sistemas biométricos.

3.3 Software

Dos aplicaciones se han usado en este estudio: MatLab y una aplicación creada para el estudio en Visual Studio.

La aplicación MatLab permitirá representar las gráficas de rendimiento. Para ello se ha tenido que utilizar una función de Biotech que se explicará más adelante.

En cuanto se refiere a la aplicación de Visual Studio se requiere que sea capaz de:

- Identificar los diferentes valores NFIQ de las imágenes de las huellas.
- Almacenar todas las imágenes sin crear problemas de memoria.
- Hacer comparativas entre las imágenes (tanto entre imágenes del mismo usuario como de usuarios distintos).
- Crear de unas listas de comparaciones genuinas y de impostores.
- Guardar estas listas en ficheros .txt para que puedan ser leidos por MatLab.

4. PROCESO DE TOMA DE DATOS

Como se ha señalado en el apartado anterior, para llevar a cabo estudios de rendimiento como los referidos en el presente trabajo, se necesita una gran base de datos. En el proceso de toma de datos se han conseguido la colaboración de casi 600 voluntarios.

Para el desarrollo de este trabajo, el GUTI permitió hacer uso de sus instalaciones a fin de facilitar el proceso de toma de datos. Por otro lado, el GUTI conservará estos datos para hacer estudios de mayor calibre y alcance.

El proceso de toma de datos consistió en dos sesiones por usuario. En cada sesión, el usuario hacía uso de tres sensores biométricos. Estos sensores se conectaban a un ordenador, el cual utilizaba una aplicación para indicar cuándo colocar el dedo en cada sensor.

4.1 Primera sesión

En la primera sesión del proceso de toma de datos, se informó inicialmente al voluntario acerca de la naturaleza y objetivos del trabajo, explicándole datos que son de su interés, tales como para qué se realiza este estudio, qué es el GUTI y cuáles son las limitaciones legales y la seguridad de confidencialidad en cuanto a la gestión de los datos personales.

Posteriormente, se le explicó al usuario el proceso, tal y como se hace a continuación, y se le informó de la contraprestación que se le ofrecía por su colaboración, consistente en esta ocasión en la entrega sin coste de dos entradas de la sala de cine CINESA. Este tipo de concesión a voluntarios es habitual en la ejecución de estudios como el realizado.

A continuación, se le entregó al voluntario un test en el que se le preguntaba sobre sus conocimientos previos y valoraciones sobre sensores biométricos.

Este test incluía, además, preguntas sobre el rango de edad y los estudios académicos del usuario para, más adelante, poder hacer un estudio de mercado sobre la visión de cada grupo social y su impresión acerca de los sensores biométricos. Este estudio no se incluye

en este trabajo, pero la obtención de los datos demográficos y sociológicos se hizo de manera simultánea, a fin de que puedan ser usados por el GUTI en estudios posteriores.

Las preguntas permitían reflejar si el usuario había tenido contacto previo con sensores biométricos, si opinaba que los sensores biométricos eran más rápidos y/o seguros que un sistema convencional de seguridad y si estaría dispuesto a usarlos en su vida cotidiana.

Todo el proceso se llevó acabo usando un software de ordenador y, en todo momento, fue el asistente de la práctica el que interactuó con el programa. Así pues, la persona colaboradora no hizo uso de él.

El software utilizado fue desarrollado por los ingenieros del GUTI. En la pantalla de inicio, el programa permitía realizar una configuración o comenzar el proceso. Cuando se escogía "realizar el proceso", el programa preguntaba sobre si se quería introducir un nuevo usuario o si se iba a trabajar con un usuario ya existente. Al escoger "nuevo usuario", el programa pedía los datos personales de este, tales como nombre, edad, teléfono de contacto, etc.

Para mayor seguridad en la fiabilidad de los datos, el asistente preguntaba al voluntario si tenía alguna característica en algún dedo que le incapacitara a este para ser empleado en el proceso de toma de datos, sea por cicatrices graves en la huella, o por lesiones o amputaciones. Una vez introducidos estos datos en el programa, si tal era caso, este no hacía pruebas con el dedo afectado.

Una vez almacenados los datos que proporcionaba el voluntario, el programa imprimía un documento de consentimiento para someterlo a la firma de colaborador.

En este documento, se incluía una descripción del proceso paso a paso, el número de visitas que tendría que hacer el voluntario, una confirmación del pago de la compensación y un acuerdo de uso de datos personales.

Este acuerdo subrayaba que ningún dato personal sería distribuido a terceros y que toda la información quedaría para uso exclusivo del GUTI. Como añadido de trasparencia y atención al usuario, el documento indicaba un correo electrónico al que este podía acudir

para plantear cualquier duda que le surgiera sobre el uso que se haría de sus datos personales.

Una vez que el usuario firmaba el documento, se le indicaba al programa que comenzaba la fase de Reclutamiento y, a continuación, el programa mostraba en pantalla las imágenes de los sensores según el orden en el que esperaba recibir los datos.

En la fase de Reclutamiento, se tomaban dos huellas dactilares válidas de seis dedos (pulgar, índice y corazón de cada mano). Para saber si una imagen era válida, primero se verificaba el NFIQ; y, si el valor era inferior a 4, se consideraba que la imagen tenía una calidad aceptable.

Como se registraban dos imágenes por cada huella, a la hora de tomar una segunda imagen el programa comparaba la segunda con la primera y comprobaba que eran iguales. De no resultar así, el programa daba la opción de volver a coger una segunda imagen o de repetir la captura de huellas para ese dedo que había dado fallo.

Si tras tres intentos no se conseguían dos imágenes válidas, el programa pasaba en todo caso al siguiente dedo, lo que permitía que se avanzara en el proceso sin repetirlo de forma indefinida. Sin embargo, cuando se daba el caso de tres intentos fallidos significaba que no se iba a tener un patrón con el trabajar en los procesos siguientes. Si no se tiene una imagen patrón cuando se intenta hacer la comparación simplemente se almacena la imagen sino tiene ningún error adicional y se sigue con el proceso.

Una vez que se habían tomado todos los patrones, comenzaba la fase de Verificación. Previo al inicio de esta fase del proceso se cambiaban de orden los sensores según indicaba el programa, esto se hace para evitar que el usuario entre en un estado monótono y realice el proceso acordándose de cómo colocar el dedo en cada momento.

En la fase de Verificación, se tomaban seis imágenes aceptadas con cada dedo utilizado en este estudio, un total de 36 patrones. En este proceso, podían aparecer dos errores: la imagen podía no tener un NFIQ suficientemente bajo y, por lo tanto, no era posible obtener unas minucias suficientes como para poder compararlas con los patrones; o el programa

podía comparar las minucias del patrón y la imagen recién tomada y rechazar la huella a pesar de ser un usuario genuino.

Independientemente de los errores que podían aparecer, se realizaban solo tres intentos. Si pasados estos tres intentos se seguía sin obtener una imagen aceptable, el proceso continuaba por los motivos antes indicados.

Una vez terminada la fase de Verificación, el programa se cerraba.

A continuación, se le indicaba al usuario que tenían pasar 15 días entre esta sesión y la siguiente, se le informaba de la fecha exacta a partir de la cual podía venir a hacer la segunda sesión y se le ofrecía reservar una hora en concreto para asegurar una adecuada organización en la atención al usuario y que los ordenadores no estuvieran siendo usados en el momento en el que fuera a volver.

Como norma general, esta primera sesión se desarrollaba en algo más de media hora:

- Los pasos previos al proceso solían requerir de entre 5 y 10 minutos.
- El Reclutamiento se procesaba en aproximadamente 5 minutos
- Y la fase de Verificación podía exigir de 15 a 20 minutos por usuario.

4.2 Segunda sesión

En la segunda sesión, se llevaba a cabo un proceso más breve que en la primera y se repetía la segunda mitad de esa sesión.

Primero, se comprobaba que el usuario no había repetido su asistencia antes de los 15 días que tenía que entre las dos sesiones.

Después, se arrancaba de nuevo el programa, indicándole que iba a trabajar con un usuario del que ya se tenían datos. Para saber qué datos debía usar, el programa pedía o bien el DNI del usuario o bien el número que le había sido asignado por el propio programa en la primera sesión.

En cuanto el programa encontraba al usuario que repetía sesión, se comenzaba de nuevo por la fase de Verificación.

Una vez más, se tomaban seis imágenes válidas por cada dedo utilizado, con tres intentos por cada toma de imagen, de tal forma que idílicamente se obtienen 36 imágenes y en el peor de los casos se deben de hacer 108 intentos (caso que no se dio en todo el proceso). Debido a que el usuario ya conocía el proceso, esta fase de Verificación solía ser más rápida que la de la primera sesión.

Al término del proceso, se le entregaba al usuario un segundo test. En este, se le pedía que calificara cada sensor según comodidad y rapidez. Después, se le volvía a preguntar su opinión sobre los sensores biométricos: si creía que son más cómodos que una contraseña, si los consideraba más rápidos y si estaría dispuesto a usarlos en su vida cotidiana.

Finalmente, se le entregaban al usuario las entradas de cine y se le pedía la firma de un documento para certificar que, en efecto, había recibido la contraprestación prometida.

Esta segunda visita no solía exceder los 15 minutos, salvo casos excepcionales. La mayoría de los usuarios la terminaba en unos 10 minutos.

5. PROCESADO DE DATOS

Para procesar los datos obtenidos, se han usado dos programas: uno, desarrollado en C# y que es específico para el estudio; y posteriormente un segundo programa desarrollado en Matlab.

El programa escrito en C# utiliza una biblioteca distribuida por NIST en la cual se incluyen una serie de funciones que permiten trabajar con imágenes y minucias.

Una vez procesado el programa, se obtienen unas listas de datos que se utilizan en Matlab. Estas listas son convertidas a las gráficas de densidad de probabilidad, ROC y DET.

5.1 Programa en C#

5.1.1. Biblioteca NIST

A fin de entender las funciones del programa, se describen a continuación la funcionalidad de las variables y las funciones de la biblioteca NIST de las que se ha hecho uso durante el trabajo [9].

Estas son las clases con las funciones utilizadas:

- *DetectMinutiae*. Esta clase se encarga de transformar imágenes en vectores minucia para que el resto de clases puedan utilizar estos vectores en sus funciones.

  La función utilizada ha sido *FromBitmap(Drawing.Bimap, image,int, pixels_per_inch)*. Esta función toma una imagen con extensión .bmp y un número que indica el número de pixeles por pulgada que contiene esa imagen. Esta función devuelve una matriz de variables de tipo *Minutia*.

- *Minutia*. Esta es la clase que forma los objetos que se utilizan durante todo el programa.

En ella se encuentran tres variables principales: la posición en *x* en donde se sitúa la minucia, la posición en *y* en donde se sitúa la minucia y, finalmente, la dirección que debía seguir el valle de no haberse dividido o acabado.

- *Matcher*. Esta clase contiene la función que más se ha utilizado a lo largo del programa: *Compare (Minutia[] probe, Minutia[] gallery)*.

  Esta función toma todas las minucias de dos huellas, las compara para comprobar si se corresponden y devuelve un número indicando el valor de similitud que tienen.

- *Nfiq*. Esta clase se ha utilizado para poder hacer una selección de calidad de imagen.

  La función utilizada ha sido *FromBitmap(Drawing.Bimap image,int pixels_per_inch)*, que emplea las mismas variables que la función en DetectMinutiae: una imagen y su tamaño. Sin embargo, en vez de devolver un grupo de minucias, simplemente da el valor NFIQ que le corresponde a esa imagen.

Viendo estas clases, se puede inferir cómo se realizó el programa: primero, se clasificó la imagen según su NFIQ; después, se almacenaron sus minucias en una matriz de minucias; y, una vez almacenadas todas las imágenes, se hicieron comparativas entre ellas utilizando la función de *Matcher*.

5.1.2 Clases Creadas

Para poder hacer un buen uso de la memoria del ordenador y comprender mejor el programa, se crearon tres clases: User, Sensor y Fingerprint (Figura 8).

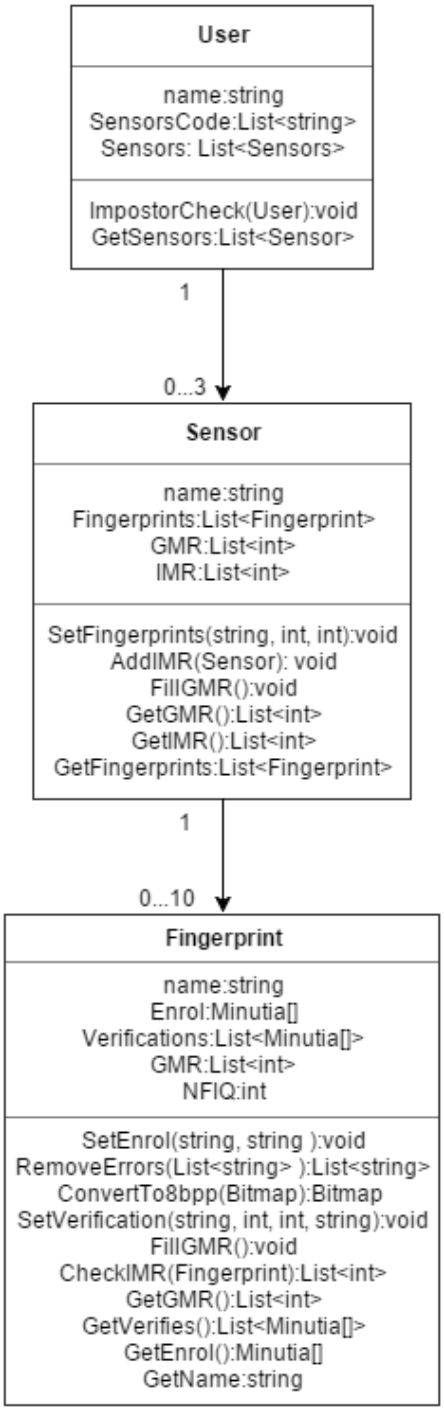


Figura 8: Diagrama de clases.

User

Esta clase crea un objeto por cada usuario utilizado en el proceso. En ella se encuentran tres variables: un nombre (*name*), una lista de los sensores que este usuario ha utilizado (*sensors*) y una lista de códigos que son asignados a cada sensor (*sensorsCode*).

El objeto *User* posee tres funciones:

- *User (string path, List<string> Code, int max, int min)*. Esta es la función de generación de objeto.

  La variable *path* indica dónde se sitúa la carpeta en la que se almacenan todas las imágenes de las huellas del usuario.

  Code se refiere a la lista de códigos de sensores usados durante el estudio.

  Las variables *max* y *min* indican el NFIQ máximo y mínimo que delimitan las imágenes válidas.

  La función genera un objeto tipo *User* al cual le asigna una lista de sensores de tamaño igual a *Code*. Estos sensores son creados los valores de *path*, *max* y *min*.

- *ImpostorCheck (User Impostor)*. Esta función es la encargada de hacer la comparativa de impostores.

  Sin embargo, en este proyecto no se estudia las comparaciones entre imágenes de distintos sensores. Por ello, esta función simplemente llama a la función *AddIMR* de cada sensor y utiliza como variable los sensores del *User Impostor*.

- *GetSensors* (). Esta función devuelve simplemente la lista de sensores asignados.

Sensor

Esta clase representa los sensores que hay dentro de cada usuario.

Esta clasificación se hace para disponer de un mejor control de que las comparativas se hacen siempre entre los mismos sensores, ya que este estudio no cubre las comparativas entre imágenes de distintos sensores.

Hay cuatro variables en esta clase: un nombre (*name*), una lista de huellas (*Fingerprints*), una lista de comparativas genuinas (GMR) y una lista de comparativas de impostores (IMR).

El objeto consta de 7 funciones:

- *Sensor (sring path, string sensorName, int max, int min)*. Esta es la función de generación de objeto.

  La variable *path* sirve, una vez más, para indicar al programa en qué carpeta ha de buscar las huellas dactilares.

  sensorName permite descartar las imágenes que no corresponden al sensor correspondiente.

  Finalmente, *max* y *min* vuelven a hacer referencia a los valores máximos y mínimos de NFIQ que han de tener las imágenes.

  En esta función se genera inicialmente una lista vacía de IMR; después se crea una lista de huellas dactilares utilizando la función *SetFingerprints* que se explica a continuación; y para concluir, se genera la lista de GMR.

- *SetFingerprints(string path, int max, int min)*. Esta función crea la lista de huellas dactilares. Como añadido, debido a que hay usuarios que pueden no haber registrado alguna huella dactilar por tener un dedo inutilizado, si después de generar una huella dactilar esta carece de patrón de reclutamiento e imágenes de verificación elimina dicha huella de la lista. Esto se hace para poder ahorrar espacio en la memoria del ordenador.

- *AddIMR(Sensor Impostor).* Esta función recibe un sensor de un usuario distinto e inicia la comparativa de los dedos de ambos usuarios.

La función revisa cada objeto de la lista *Fingerprints* y comprueba si contiene un patrón de reclutamiento. Si este es el caso, llama a la función *CheckIMR* del objeto *Fingerprint* para iniciar la comparativa de todos los dedos del "impostor". Este proceso se repite para todas las huellas de la lista del usuario.

- *FillGMR*(). Esta función recibe la lista GMR de cada una de lass huellas del usuario y las junta en una única lista. Esta lista se guardará como ".txt".

- *GetGMR*(). Devuelve la lista GMR del usuario.

- *GetIMR*(). Devuelve la lista IMR del usuario.

- *GetFingerpints*(). Devuelve la lista *Fingerprints* del usuario.

Fingerprint

Fingerprint es, sin duda, la clase más compleja, puesto que en ella se realizan los cálculos que se han ido pasando objeto tras objeto.

Tiene cuatro variables: un nombre (*name*), un *array* de minucias que corresponden a su patrón (*Enrol*), una lista de *arrays* de minucias correspondiente a sus imágenes de verificación (*Verifications*) y una lista de comparaciones genuinas (GMR).

La clase posee las siguientes 11 funciones:

- *Fingerprint (string path, string Sensor, string Fingername, int max, int min).* Es la función de generación de objeto.

  Asigna *name* con *Fingername*, utiliza *path* como método para saber en qué directorio buscar las huellas y *Sensor*, *max* y *min* para delimitar las imágenes a seleccionar.

  Lo primero que hace es asignar el *Enrol* del objeto con la función *SetEntol* y la lista de *Verifications* con la función *SetVerifications*, que se explican más adelante. Si la huella posee un patrón de reclutamiento, entonces genera la lista GMR con la función *FillGMR* que se explica también a continuación.

- *RemoveErrors (List<string> FilesPath).* Esta función revisa la lista de imágenes *FilesPath* y, buscando en sus nombres, comprueba si alguno tiene un código de error, como FTA (*Fail to Adquire*) o FTE (*Fail to Enrol*).

  Si alguna de las imágenes tiene algún error, la elimina de la lista y devuelve esta con los datos eliminados.

- *ConvertTo8bpp (Bitmap old bmp)*. Esta función se encarga de convertir una imagen de un formato 32 bpp *coloured* y transformarla en un formato 8 bpp *indexed*.

  La función es necesaria, puesto que las imágenes de las huellas están guardadas con el formato 32 bpp, pero la función Compare utiliza imágenes de formato 8 bpp.

  Es importante destacar que esta función no ha sido desarrollada por el autor de este TFG sino que ha sido obtenida de [10].

- *SetEnrol(string path, string Sensor)*. Esta función es la encargada de escoger el patrón de reclutamiento que representa al dedo concreto con el sensor específico.

  La función busca todas las imágenes de reclutamiento que corresponden a este sensor y a este dedo; elimina de la lista aquellas imágenes que contienen un error con la función *RemoveErrors*; a continuación, transforma todas las imágenes al formato deseado con la función *ConvertTo8bpp* y compara sus valores de NFIQ; finalmente, escoge el patrón con NFIQ más bajo y se lo asigna a este dedo.

- *SetVerifications(string path, int max, int min, string Sensor)*. Es la función que consigue todas las imágenes de verificación del dedo.

  De forma similar a *SetEnrol*, *SetVerification* busca todas las imágenes correspondientes a las verificaciones del sensor y dedo específicos.

Una vez obtenida esta lista, descarta cualquier huella con NFIQ superior a *max* o un NFIQ menor a *min*.

Finalmente, asigna esta lista de huellas a *Verifications*.

- *FillGMR()*. Es la función que asigna los valores de GMR utilizando la función *Compare* con todas las verificaciones y el patrón de reclutamiento.

- *CheckIMR(Fingerprint Impostor)*. Esta función compara las verificaciones del impostor con el *enrol* del dedo genuino. Estos valores se añaden a la lista IMR del usuario genuino.

- *GetGMR()*. Devuelve la lista de comparaciones genuinas GMR.

- *GetVreifies()*. Devuelve la lista de verificaciones *Verificactions*.

- *GetEnrol()*. Devuelve el patrón de reclutamiento *Enrol*.

- *Getname()*. Devuelve el nombre del dedo *name*.

5.1.3 Flujo del programa

El programa creado tiene tres bloques diferenciados (figura 9):

- *Creación de datos*. En esta parte del proceso se crean todos los objetos necesarios.

  Del total de usuarios, cada usuario se compone de tres sensores y cada sensor de un máximo de seis dedos. Esto permite poder distinguir entre distintos sensores y huellas correspondientes a esos sensores.

  En este proceso, también se asignan las listas de comparaciones genuinas GMR, de tal forma que cada sensor de cada usuario tiene una lista GMR asignada.

- *Comparaciones de Impostores*. En esta fase del programa se hacen las comparaciones de impostores.

Para ello, se selecciona un usuario y se compara su patrón de reclutamiento con todas las imágenes de verificación del resto de los usuarios. Este proceso se hace para cada usuario.

– *Guardado de datos.* Todas las listas de GMR de cada sensor se guardan en un archivo .txt y, de forma similar, se hace la misma operación con las listas de IMR, de tal forma que, al final se obtienen una lista GMR y una lista IMR por cada sensor que se ha utilizado.

  Este proceso se repite para los distintos grupos de calidad de NFIQ.

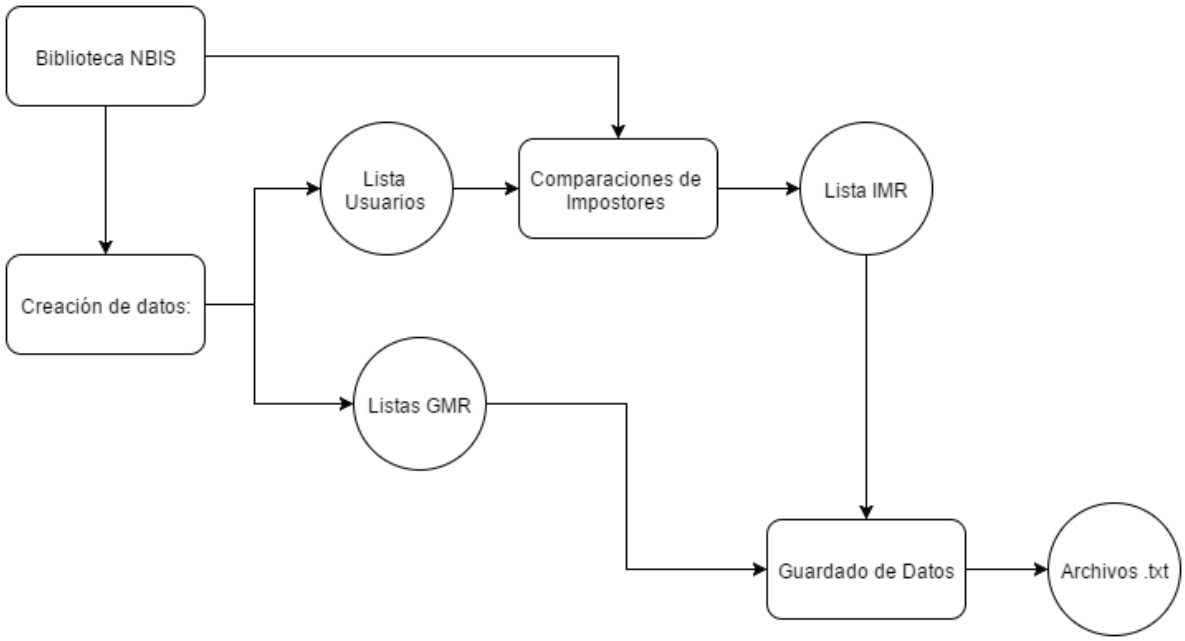


Figura 9: Diagrama de flujo general del programa C#

5.1.4 Creación de datos

Todos los datos recopilados durante el proceso de toma de datos se guardan en un carpeta llamada Muestras.

Dentro de ella, se encuentra una carpeta por cada usuario. La carpeta de cada usuario tiene el código de ese mismo usuario. Por ejemplo, todas las imágenes del usuario 100105 se

encuentran en la carpeta C:/…/Muestra/100105; y, en ella, se hallan todas las imágenes de cada sensor, ya sean de verificación o de reclutamiento.

Para trabajar con estas imágenes, el programa necesita primero hacer una lista de usuarios con cada sensor y huella. Para ello, se crea una lista de carpetas que indica el número de usuarios que existen. Una vez creada la lista, cada usuario es sometido a los siguientes procesos (figura 10):

- Se buscan las imágenes correspondientes a un sensor y se crea un objeto sensor que pertenece al usuario.

- Dentro de cada sensor, se busca cada dedo que tenga imágenes de ese sensor.

- Para cada dedo, se escoge una única imagen de reclutamiento (la de menor NFIQ) y se añaden las verificaciones a la lista *Verifications*.

- Una vez que se crea la lista y *Enrol[]*, se hace una comparación entre el *Enrol* y cada verificación, añadiendo el valor de la comparación a la lista GMR del dedo.

- Este proceso se sigue por cada dedo de cada sensor; se repite después con todos los sensores de cada usuario, y finalmente con cada usuario existente en la base de datos.

Cabe mencionar que, si un dedo carece de *Enrol*, el proceso de comparación genuina no se realiza; y si un dedo no tiene ni *Enrol* ni *Verifications*, es eliminado de la lista dedos para evitar comparaciones innecesarias.

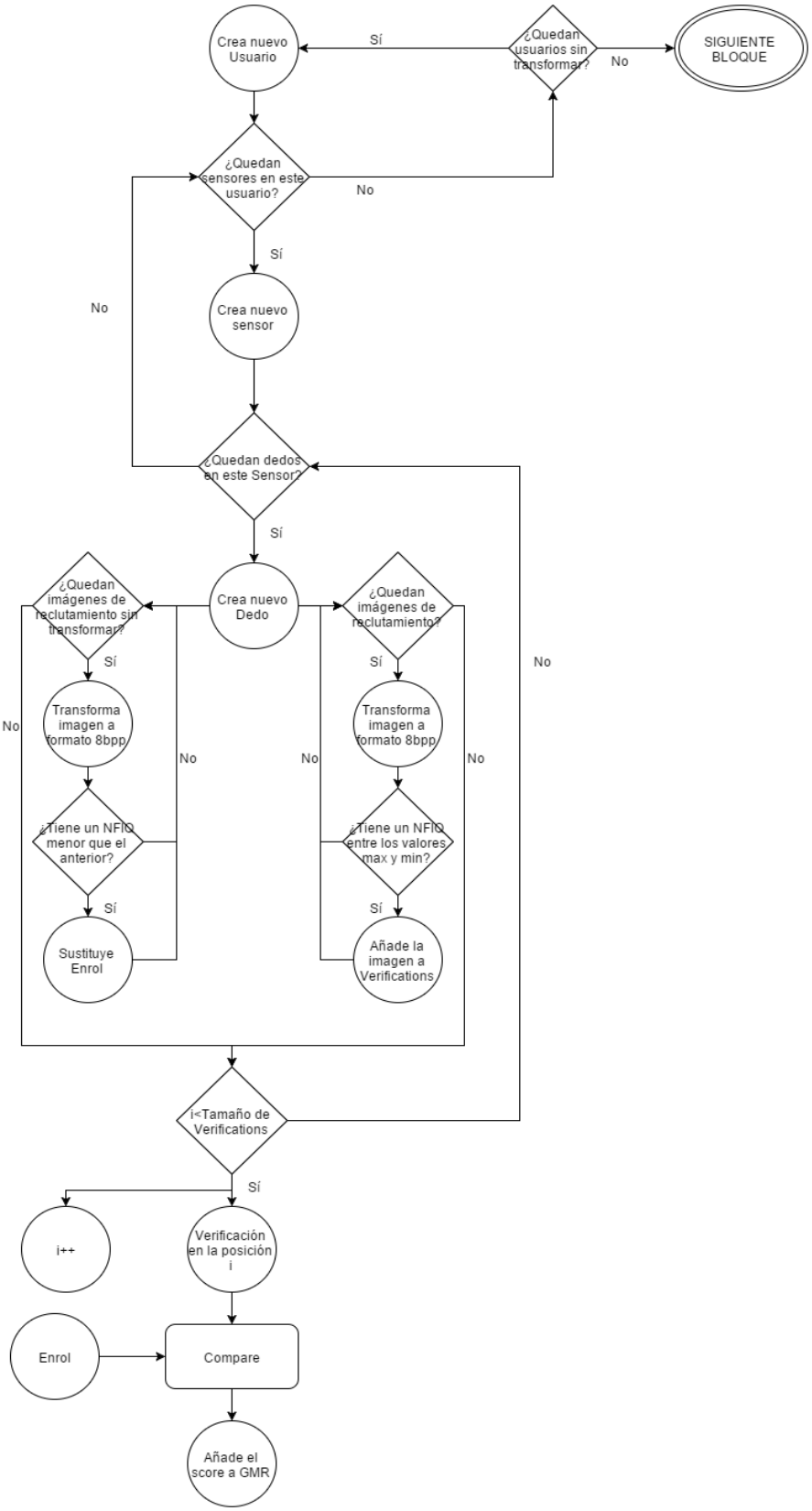


Figura 10: Diagrama de flujo del bloque de creación de datos.

5.1.5 Comparación de “impostores”

Esta es la parte del proceso que hace que la ejecución del programa requiera varias horas (figura 11).

Para hacer una lista de comparaciones con impostores, hay que tener en cuenta que cualifica como comparación de impostor no sólo la comparación de un dedo del impostor con su correspondiente dedo para otros usuarios, es decir, del dedo índice de un usuario con el dedo índice de otro usuario, sino que también se tiene que hacer la comparación del dedo índice con el dedo pulgar y corazón de ambas manos. Se considera todas estas comparaciones como impostoras por la idea de que un impostor podría intentar probar todas sus huellas para acceder al sistema. Esta comparación se hace con todas las imágenes de *enrol* del impostor. Esto explica que la mayor parte del tiempo de uso del programa se dedique a este proceso.

Sin embargo, el proceso es relativamente sencillo de entender. Se escoge por orden un usuario al que se llamará *genuino* y otro usuario al que se llamará *impostor*. Estos usuarios no serán en ningún caso el mismo.

Una vez escogidos ambos usuarios, se selecciona un sensor y, de ese sensor, un dedo del usuario *genuino*, que se pasará por todos los dedos del usuario *impostor* con el sensor adecuado. Seguidamente, se hace la comparación de todas las imágenes de verificación del usuario *impostor* con el usuario *genuino* y los valores de estas comparaciones se guardan en la lista IMR del sensor del usuario *genuino*.

Cuando se termina con estas comparaciones, se repite el proceso con los otros sensores.

Tras haber realizado el proceso con todos los sensores, se escoge un nuevo usuario *impostor* y, una vez realizado el proceso con todos los *impostores*, se pasa a un nuevo usuario *genuino* y se repite todo el proceso anterior.

Para hacerse una idea del número de comparaciones que se han de realizar, un sensor puede tener unas 2.000 comparaciones en su lista GMR y unas 400.000 comparaciones en su lista IMR por cada grupo de calidad.

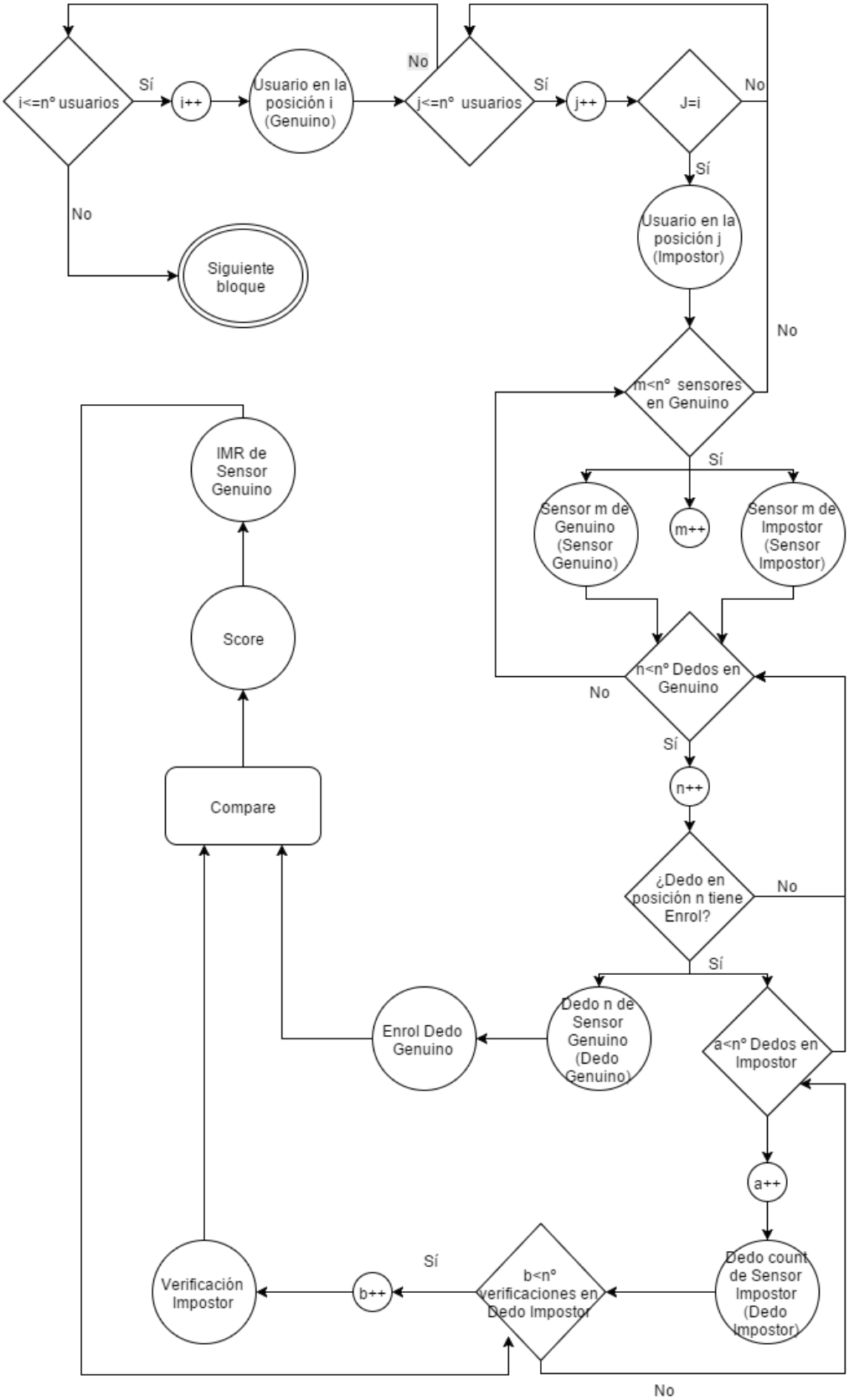


Figura 11: Diagrama de flujo del bloque de comparación de impostores.

5.1.6 Almacenamiento de los datos

Los datos obtenidos de este programa se procesaron posteriormente en Matlab para poder obtener las medidas de rendimiento.

A la hora de representar valores en Matlab, es más sencillo enviar una lista de densidad en vez de una lista de valores. Esto quiere decir que es más sencillo leer un archivo que diga "hay dos comparaciones con valor 0, cuatro con valor 1, etc.", que uno que vaya dando valores de forma aleatoria.

Para poder guardar de la forma adecuada los valores, se sigue el siguiente proceso (figura 12):

- Primero se escoge el sensor a utilizar y se crea un archivo .txt con el nombre del sensor, la calidad correspondiente y la lista usada (ej: NXT5-4GMR.txt").

- Una vez creado el archivo, se comienza a llenar la lista GMR genérica. Para poder llenarla lo que se hace es ir a las listas GMR correspondientes se leen los valores de la lista y se añade un 1 a la posición de la lista GMR genérica.

A título de ejemplo, imagínese una lista GMR genérica nueva que carece de valores. Al leer el primer valor de una lista GMR, este corresponde con un *Score* de 0. En este caso, se añade un 1 a la posición 0 de la lista genérica. Si el valor del *Score* es mayor que la lista genérica, entonces se añaden 0's hasta llegar a ese valor.

En el ejemplo anterior, si el siguiente valor corresponde a un *score* de 4, el programa añade un 0 en la posición 2ª y 3ª de la lista genérica y un 1 en la 4ª posición.

Una vez llenada la lista GMR, se repite el proceso con las listas IMR.

Cuando todos los valores se han almacenado en las listas genéricas, se guardan los valores en los archivos .txt y se borran los datos almacenados en las listas para volver a empezar el proceso con un nuevo sensor.

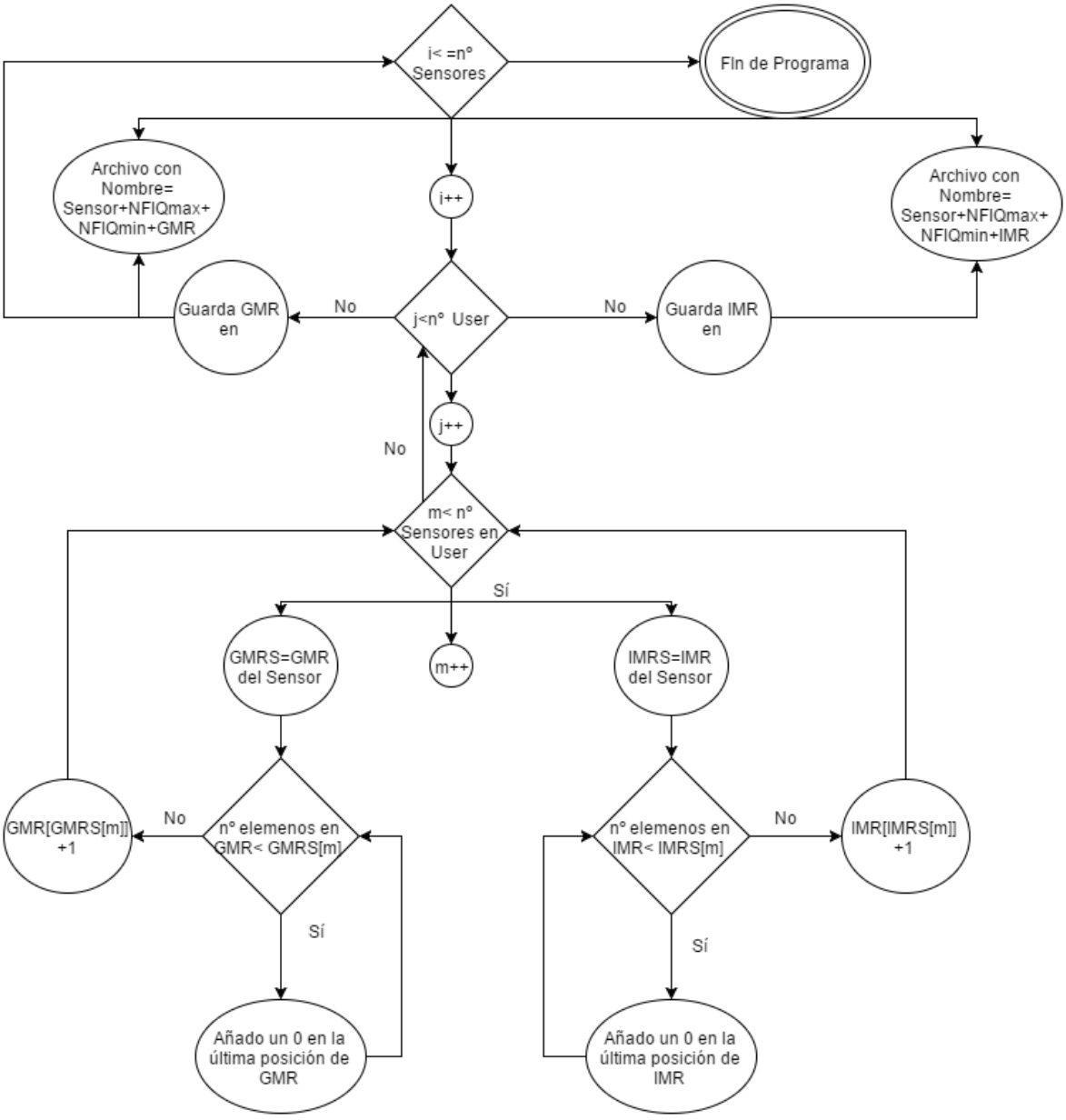


Figura 12: Diagrama de flujo del bloque de guardado de datos.

5.2 Matlab

Una vez terminados los distintos procesos del programa en C#, se acaba obteniendo una serie de archivos con listas de datos.

Mirando simplemente estos datos, resultaría muy difícil determinar los rendimientos de los sensores. Afortunadamente, la compañía Biotech ha creado una biblioteca de Matlab, la

cual permite que, con una lista de datos, se puedan crear las gráficas ROC, DET y FARvsFRR.

Además, resulta muy sencillo hacer la gráfica de densidad de probabilidad en Matlab. Para ello, simplemente se crean dos vectores: GMR y IMR. Con la función *importdata*, se pasan los archivos correspondientes a los dos vectores y, antes de representar los vectores, se dividen todos los datos por el valor máximo, de tal forma que se normalizan los resultados.

La normalización se realiza debido a que el vector IMR suele tener entre 100 y 200 veces más datos que la función GMR, lo cual hace que, a la hora de representarlos, la función GMR apenas pueda verse.

En cuanto los datos han sido normalizados, se crean las gráficas con la función *plot*.

Una vez representada la curva de densidad, se usa la función EER_DET, que representa las otras tres gráficas de forma automática.

Esta función pide cuatro valores:

- La lista de usuarios genuinos.
- La lista de usuarios impostores.
- Un valor de aproximación de valor óptimo
- El número de valores umbral que se quiere utilizar.

El valor óptimo se refiere al valor porcentual de FAR que se considera óptimo para el sistema que se quiere crear, en otras palabras el porcentaje de usuarios impostores el cual es permisible que accedan al sistema(como no se está creando ningún sistema en concreto, este valor no es influyente). Se suele aconsejar que el número de valores umbral sea de 10.000, así que se ha usado ese valor.

Este proceso se repite con todos los sensores y calidades, a fin de disponer de suficientes gráficas para poder hacer las distintas comparativas.

6. RESULTADOS

Una vez obtenidas las gráficas ya se puede analizar la influencia de los niveles de calidad en los sensores utilizados. Los resultados obtenidos entran dentro de lo esperado pero sí que se pueden encontrar algunas sorpresas al ver las calidades media y alta.

Para poder entender mejor los resultados, estos se han dividido en distintos capítulos. Primero se analizará cada sensor por separado, comparando los distintos niveles de calidad entre ellos. Y en el último apartado se comparará todos los sensores a distintos niveles de calidad. Esto nos permitirá ver la influencia de la calidad en cada sensor y ver la comparación de rendimientos de mejor forma.

6.2 NXT

Observando el sensor NXT podemos ver que dependiendo del valor umbral que se quiera escoger los valores de calidad media y calidad alta pueden llegar a ser iguales.

En una primera inspección de los resultados, se puede ver la gran diferencia entre la calidad más baja con el resto de calidades obtenidas. Observando el DET (figura 13) podemos ver como para un porcentaje de FAR la diferencia entre la calidad baja con el resto es de gran significado. Cogiendo, por ejemplo, un 5% máximo de FAR las calidades alta y media permitirían un FRR de aproximadamente un 20% pero, en el caso de la calidad baja el FRR es superior al 40%. Otra clara evidencia de la diferencia entre la calidad baja y el resto de calidades se ve en el ROC (figura 14), donde las calidades alta y media pueden permitir un 50% de usuarios asegurando que casi ningún usuario sería impostor, por el contrario, la calidad baja tendría un FAR de como mínimo un 5% al dejar pasar más del 50% de usuarios. Finalmente, otro claro ejemplo de la diferencia entre estas calidades se encuentra en la gráfica FARvsFRR (figura 15), en ella se ve que si se quiere tener un FRR de casi 0% los valores de calidad alta y media tendría una tasa FAR de 40% y 60% respectivamente mientras que la calidad baja ya supera el 90% FAR en este caso.

Unos datos más sorprendentes se encuentran en las diferencias entre las calidades media y alta. Vemos que en muchos casos la diferencia entre estas calidades es muy baja llegando

incluso a cortarse. Fijándose de nuevo en la DET (figura 13) se puede ver que para tasa FAR más altas, las tasas FRR de ambas calidades son prácticamente iguales, llegando incluso a ser mejores las tasas de la calidad media. En la gráfica ROC vemos resultados similares (figura 14), las tasas FAR de ambas calidades cada vez tener resultados más parecidos hasta llegar al punto en que al llegar a porcentajes muy altos de usuarios aceptados la FAR de la calidad media es mayor. En cuanto a diferencias a la variación de las tasas FAR y FRR con respecto al valor umbral escogido se observa resultados más esperados, la calidad alta es superior en este campo que la calidad media de forma significativa.

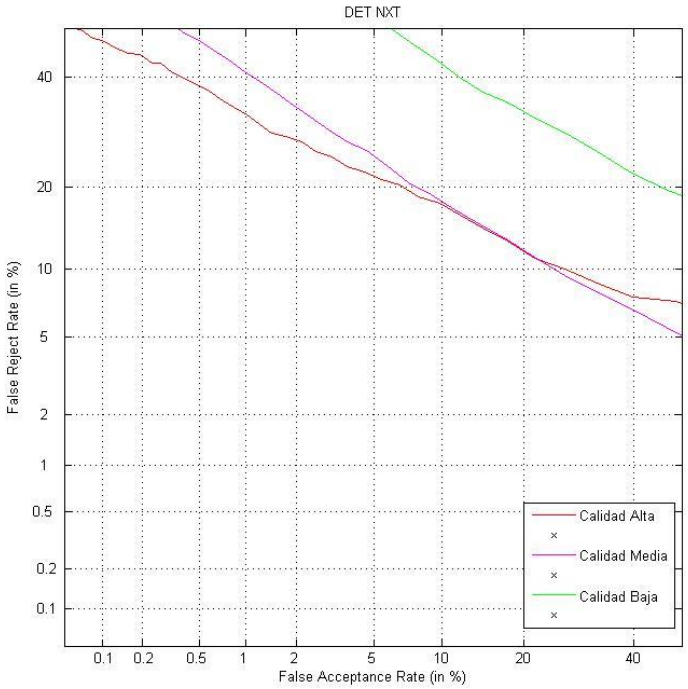


Figura 13: DET del sensor NXT.

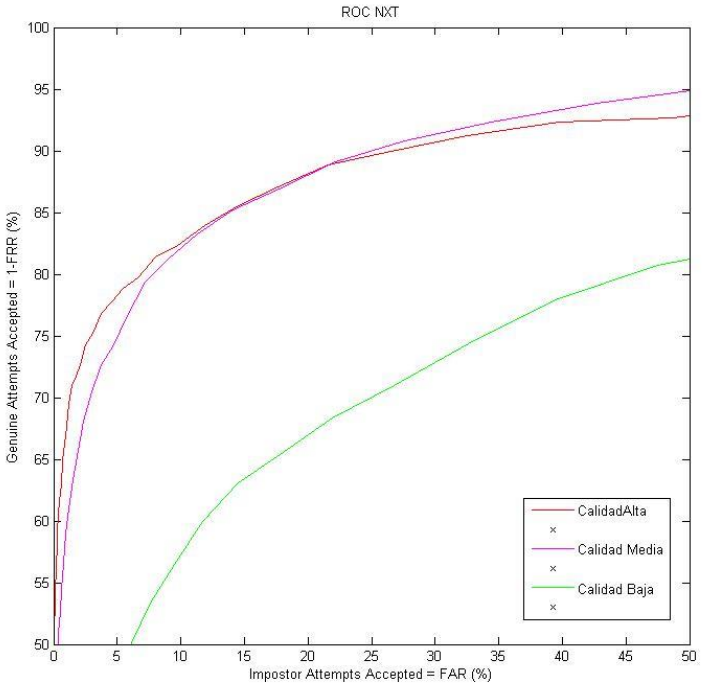


Figura 14: Gráfica ROC del sensor NXT.

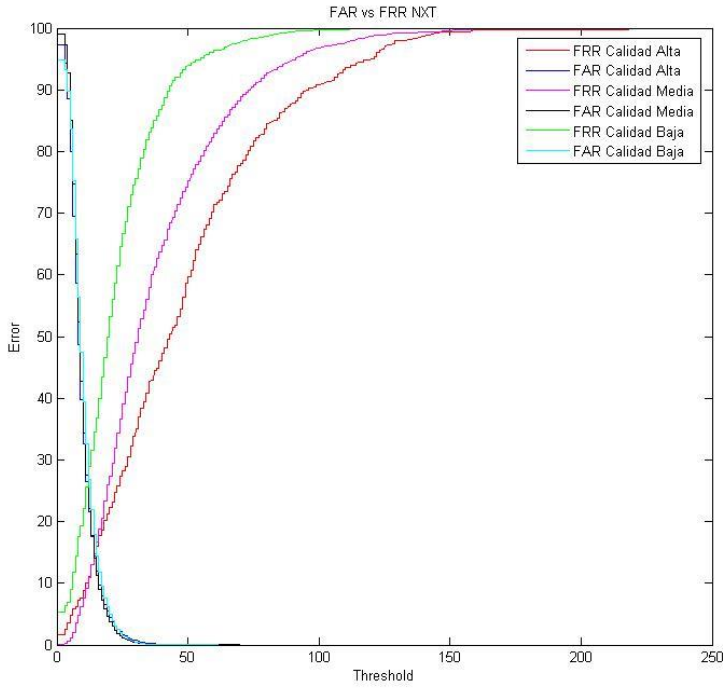


Figura 15: Gráfica FARvsFRR del sensor NXT.

6.3 FPC

Con respecto al sensor FPC se han obtenido resultados más esperados: la calidad alta, media y baja mantienen una diferencia clara y más o menos constante.

Las proporciones de la tasa FRR con respecto a la tasa FAR son claramente distinguibles entre las tres calidades (figura 16). Llama algo más la atención que la diferencia entre la calidad media y la calidad alta es menor que la diferencia entre la media y la baja. También es claro como la calidad baja tiene un rendimiento muy bajo, para que la tasa FAR empiece a ser menor del 50% se tiene que permitir una FAR superior al 10%, mientras que las calidades media y alta ya eliminan ese 60% de usuarios impostores a cambio de rechazar menos del 1% de usuarios genuinos.

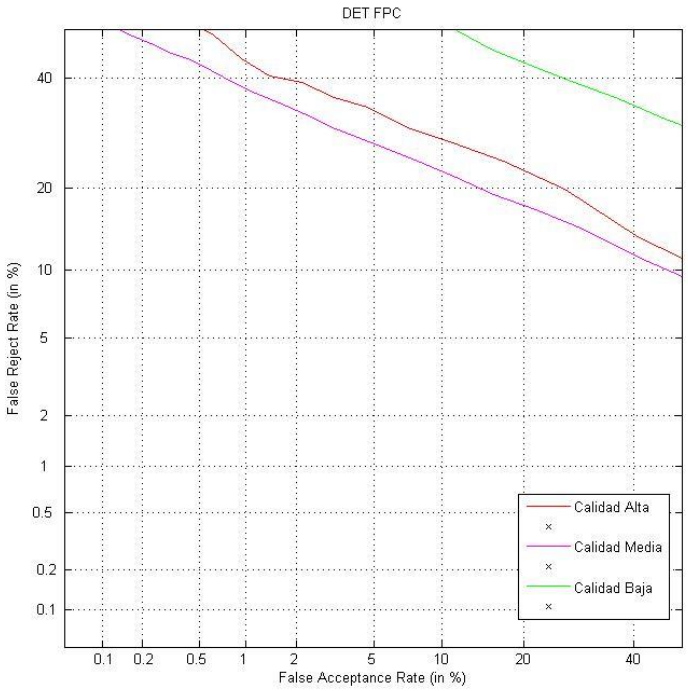


Figura 16: DET del sensor FPC.

En el ROC vemos resultados similares a la gráfica DET (figura 17): gran diferencia entre la calidad baja y media, diferencia de resultados más o menos constante y un rendimiento de la calidad baja significativamente menor que el resto de calidades. Sí cabe mencionar que

las calidades media y baja tienen resultados muy similares para porcentajes de usuarios por debajo del 50%.

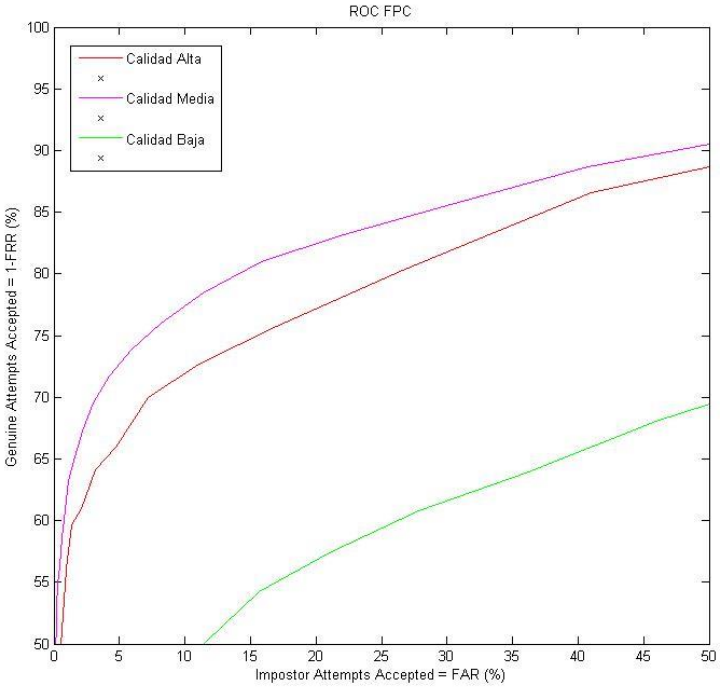


Figura 17: Gráfica ROC del sensor FPC

Finalmente en la gráfica FARvsFRR sorprende ver como la influencia del valor umbral es más similar entre las calidades baja y media que en las gráficas anteriores (figura 18). Para umbrales bajos se percibe una mayor similitud entre los valores de Error FAR de la calidad media y baja que con la calidad alta, a esto se le añade el hecho de que ambas calidades pasado el umbral de 50 se vuelve difícil distinguirlas debido a que muestran resultados muy similares.

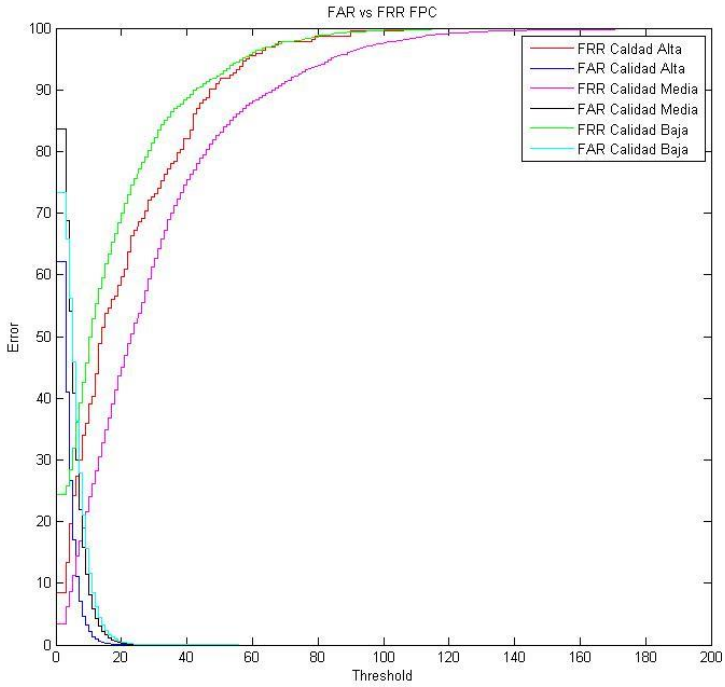


Figura 18: Gráfica FARvsFRR del sensor FPC.

6.3 UPK

Al igual que el sensor FPC, el sensor UPK tiene en mayor parte bastante distinguible las calidades alta, media y baja.

Si se estudia la gráfica DET es fácil ver que los resultados son claramente diferenciables (figura 19). Una vez más, la diferencia entre la calidad media y la calidad alta es muy baja en comparación con la calidad baja. En este sensor también se puede ver cómo el rendimiento de la calidad baja es muy bajo con proporciones de más del 40% de FRR para una FAR menor del 20%.

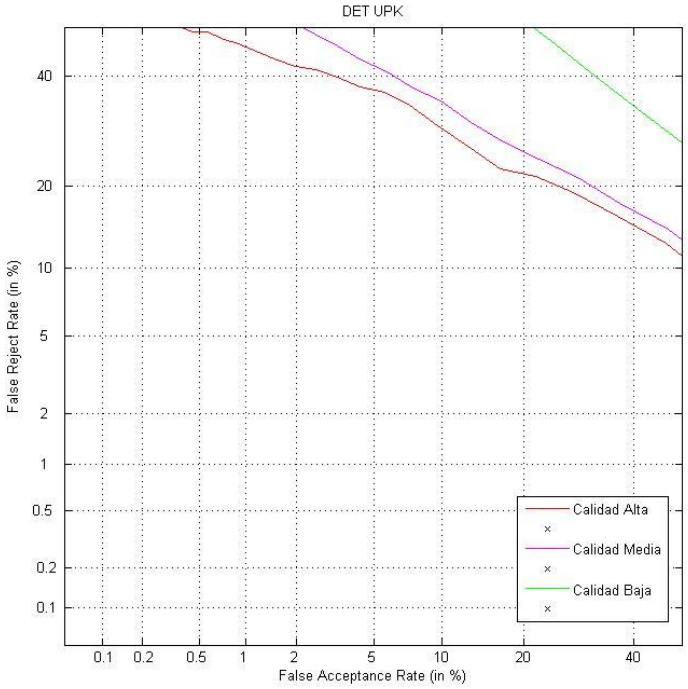


Figura 19: DET del sensor UPK.

Analizando la gráfica ROC es claro que los resultados alcanzados con este sensor son más distinguibles que con el resto de sensores (figura 20). Al contrario que en el resto de sensores, la calidad media y alta ya son algo diferentes para un porcentaje de aceptación del 50% de usuarios. Sin embargo, una constante que se mantiene es la clara diferencia entre la calidad baja y el resto de calidades.

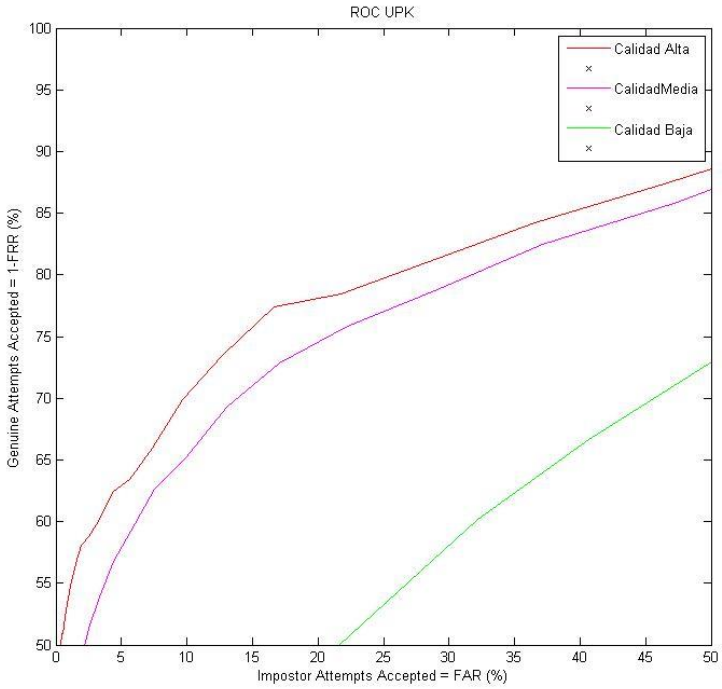


Figura 20: Gráfica ROC del sensor UPK.

Llegados a la gráfica FARvsFRR se puede observar un cambio de papeles entre la calidad media y alta (figura 21). Al contrario que en el resto de sensores donde la gráfica FARvsFRR suele mostrar resultados más esperados, en el caso del sensor UPK hay que fijarse en que la calidad media parece reaccionar mejor a valores umbral más alto que la calidad alta.

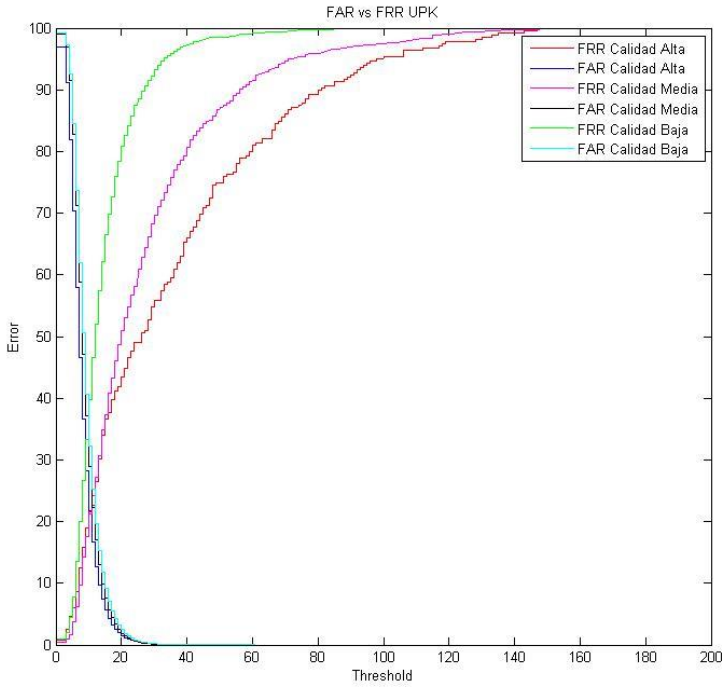


Figura 21: Gráfica FARvsFRR del sensor UPK.

6.4 Comparativa entre sensores

Finalmente se ha hecho una comparativa de los tres sensores a sus distintas niveles de calidad para comprobar cual tiene mejor funcionamiento dependiendo del nivel de exigencia del sensor.

Fijándonos en las gráficas que representan los valores de alta calidad vemos grandes diferencias entre algunos de los sensores (figuras 22, 23 y 24). Los sensores UPK y FPC representan valores muy similares, exceptuando en la gráfica FARvsFRR donde se puede ver que el sensor UPK es ligeramente mejor que el sensor FPC. A pesar de esto lo sorprendente es el sensor NXT, el cual demuestra tener un rendimiento bastante mejor que el de sus compañeros.

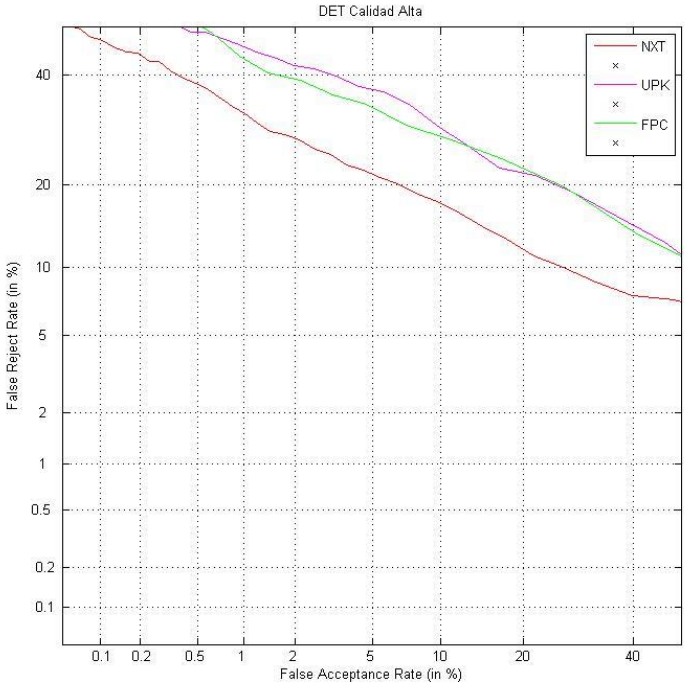


Figura 22: DET de los sensores a calidad Alta.

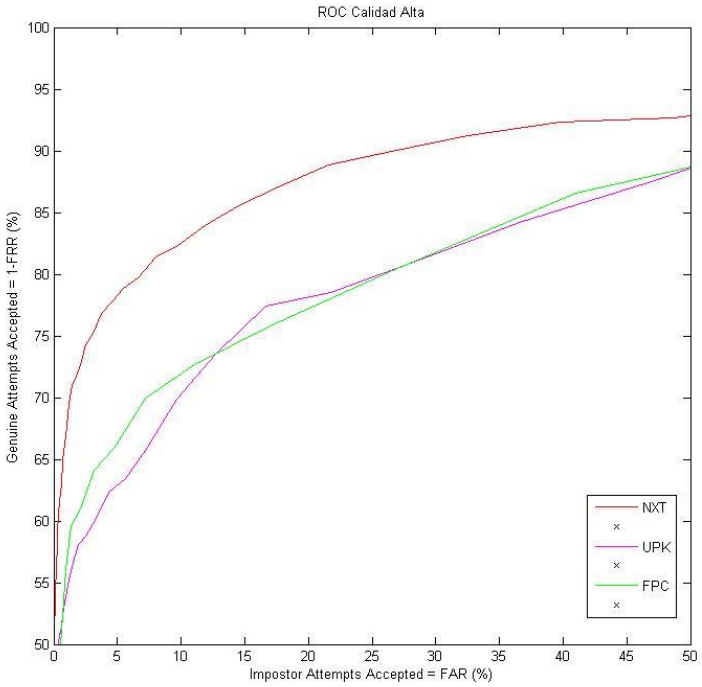


Figura 23: ROC de los sensores a calidad Alta.

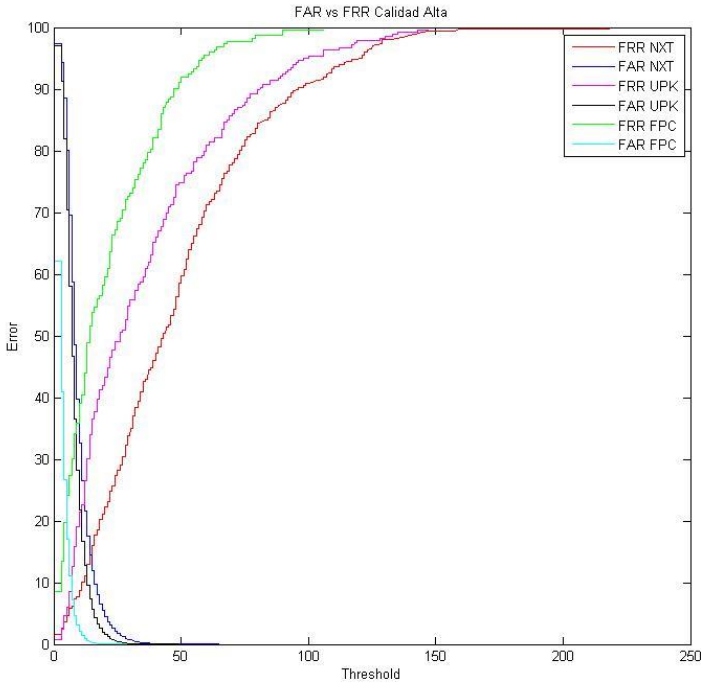


Figura 24: FARvsFRR de los sensores a calidad Alta.

Pasando a la calidad media se puede ver una mayor aproximación de los sensores (figuras 25, 26 y 27). El sensor FPC y NXT obtienen valores similares para grupos de personas pequeños y umbrales poco exigentes. Dónde sigue existiendo grandes diferencias es en el sensor UPK. Este sensor sigue mostrando unas prestaciones visiblemente más bajas que los otros dos sensores.

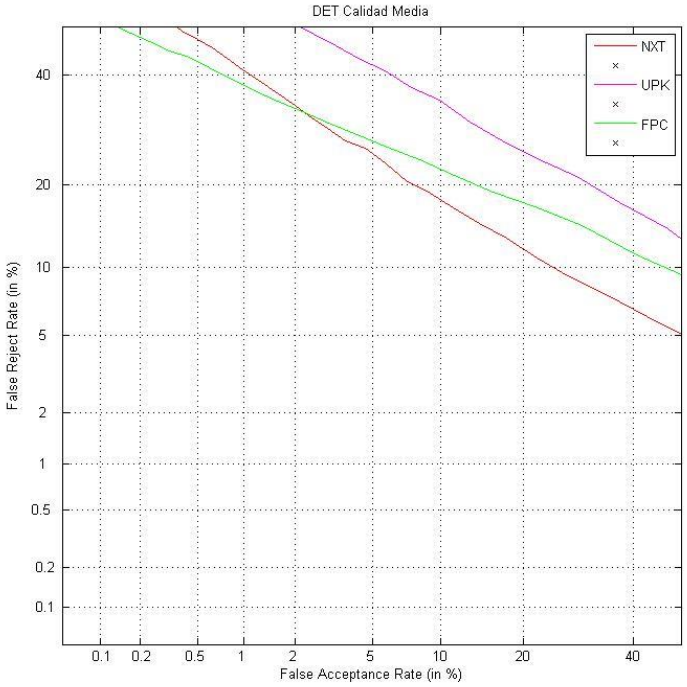


Figura 25: DET de los sensores a calidad Media.

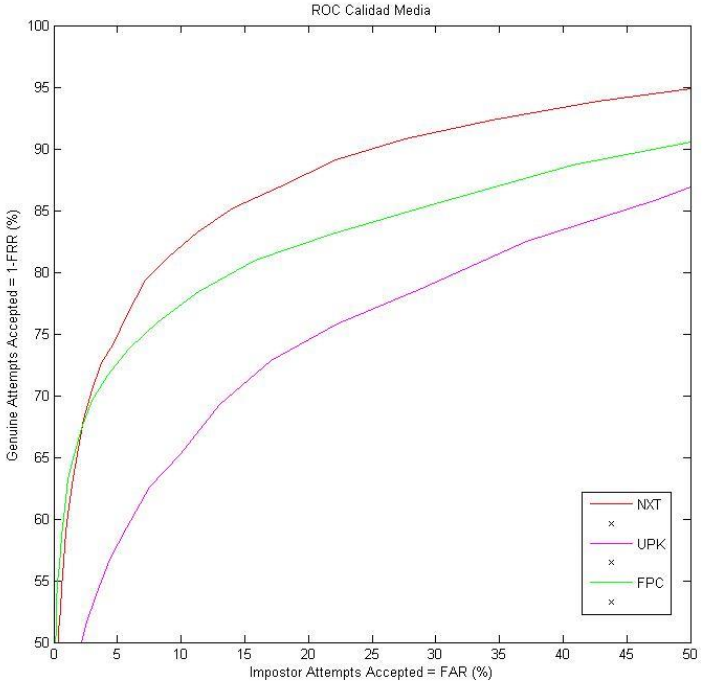


Figura 26: ROC de los sensores a calidad Media.

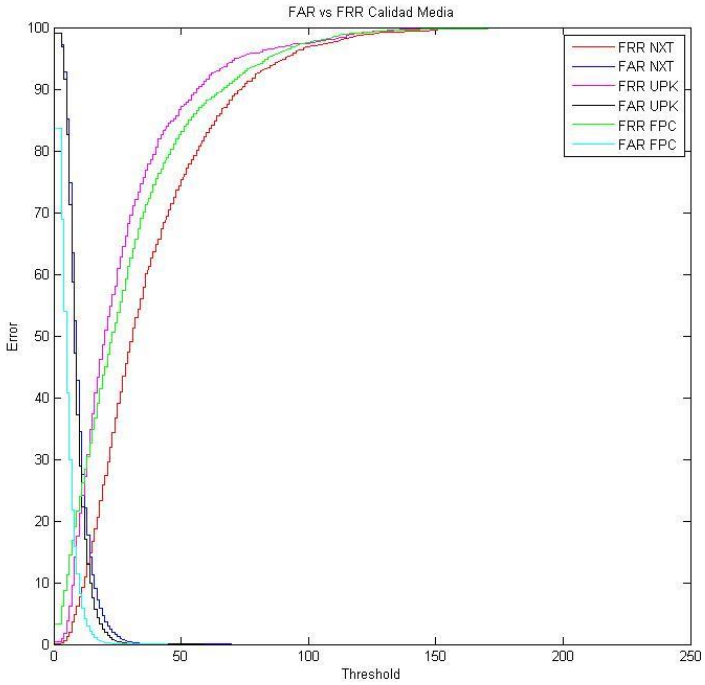


Figura 27: FARvsFRR de los sensores a calidad Media.

Llegados a los valores de NFIQ más altos se ve que los sensores dejan de tener valores aceptables (figuras 28, 29 y 30). Viendo la gráfica DET se ve que para eliminar cerca de un 80% de usuarios no deseados se rechazaría también entre un 30% y un 50% de usuarios genuinos. Si se observa la gráfica FARvsFRR se ve que hay poco margen a la hora de aumentar el valor umbral debido a que los valores de FRR crecen demasiado rápido.

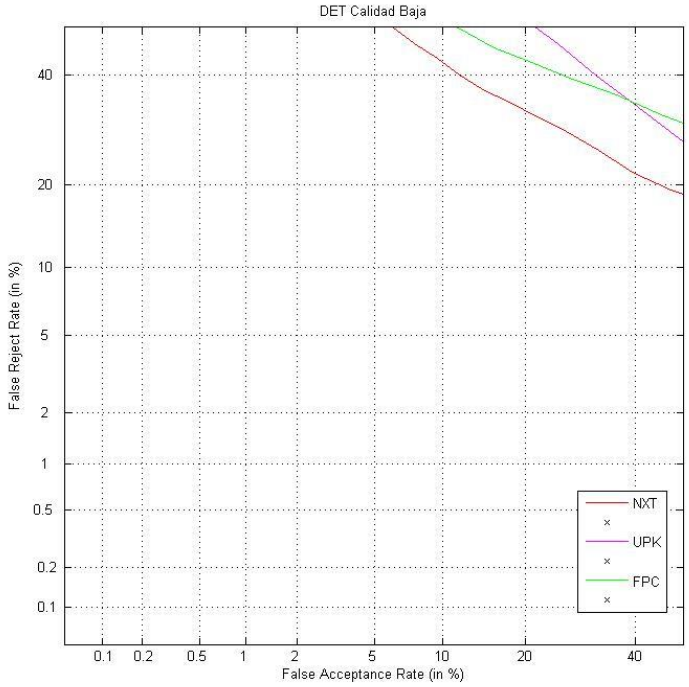


Figura 28: DET de los sensores a calidad Baja.

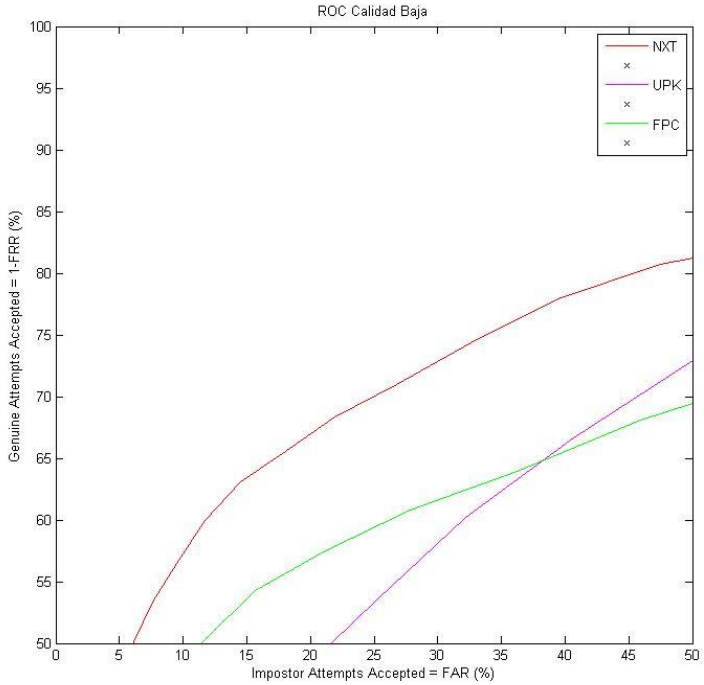


Figura 29: ROC de los sensores a calidad Baja.

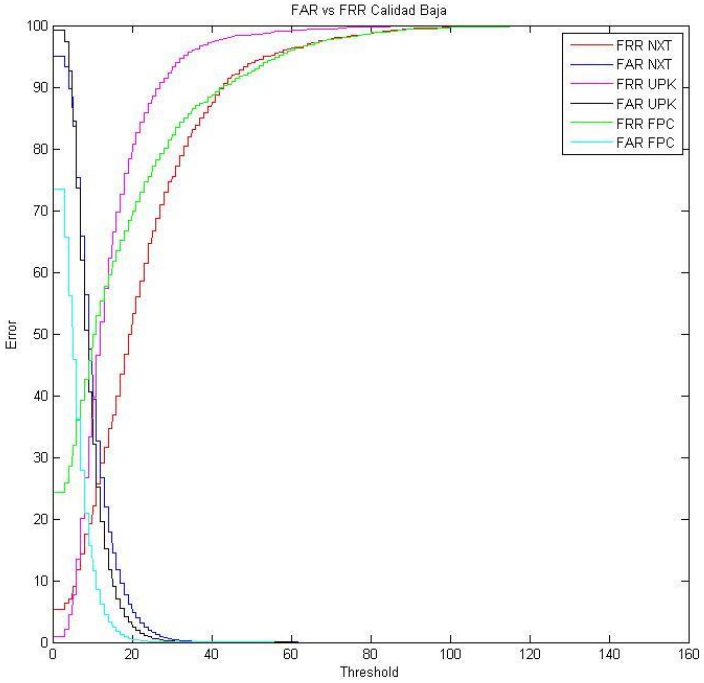


Figura 30: FARvsFRR de los sensores a calidad Baja.

7. CONCLUSIONES

Con los resultados obtenidos, ya se puede llegar a una serie de conclusiones que respondan a los objetivos propuestos al inicio de este estudio.

El análisis realizado quería comprobar como los valores del algoritmo NFIQ influyen realmente en los rendimientos de los sensores biométricos. Este algoritmo es una primera aproximación a las calidades de las huellas, a pesar de ello es importante comprobar si estos valores realmente influyen en el rendimiento de los sensores.

Gracias al proyecto llevado se ha conseguido crear una base de datos de casi 600 usuarios. Los datos obtenidos gracias a esto usuarios podrán ser utilizados en futuros estudios relacionados con el rendimiento de los tres sensores utilizados.

También, gracias a poder haber utilizado tres sensores de tres fabricantes distintos, se puede establecer una comparativa de los sensores. Esta comparativa no tan sólo permite ver qué sensor tiene mejor rendimiento que otro, al mismo tiempo ayuda a verificar que los valores NFIQ influyen de la misma forma a todos los sensores biométricos.

7.1 Influencia de la calidad sobre el sistema

El trabajo realizado muestra algunos datos interesantes sobre la diferencia entre las calidades de los valores NFIQ establecidos por el grupo NIST. Para entender mejor los resultados vamos a revisarlos calidad por calidad:

- Calidad Baja: Vemos que los valores envueltos entre los NFIQ 4 y 5 dan unos resultados de un rendimiento muy bajo. Esto no resulta especialmente sorprendente debido a que ya NIST advierte que valores de esta calidad de imagen no se deben utilizar debido a sus bajas prestaciones. Este estudio no ha hecho más que confirmar lo establecido por NIST. Resultados que indican que si se exige que haya menos de un 10% de impostores se tienen que perder más del 40% de usuarios genuino demuestran que cualquier otro sistema de seguridad

por código o tarjeta de identificación pueden resultar extremadamente más eficientes que un sistema basado en huella dactilar.

- Calidad Media: Ha este nivel se encuentran resultados más esperanzadores. Con una exigencia de un NFIQ de 3 se puede ver que ahora eliminar un 10% de impostores conlleva una pérdida de entre el 15% y el 30% de usuarios genuinos. Estos valores son más aceptables. Cabe mencionar que estos valores además pueden ser más exigentes debido a que la diferencia entre los valores FRR y FAR permiten escoger un valor umbral más alto sin perder tantos usuarios. A este nivel se puede llegar a competir con otros sistemas de seguridad más clásicos.

- Calidad Alta: En los niveles de NFIQ 2-1 se ven resultados muy similares a los de calidad media. La mayor diferencia se encuentra para umbrales de decisión más altas, donde las curvas se suavizan más rápidamente. Esto indica que para sistemas en el que se pueden permitir un 10% de usuarios impostores la diferencia de rendimiento con respecto a la Calidad Media es mínima. Sin embargo, donde sí se ve más diferencia es cuando el nivel de exigencia es notablemente más alto, para niveles de exigencia en el que se permite menos de un 5% de impostores los NFIQ de 2 ó 1 demuestran una mayor aceptación de usuarios genuinos y un mayor porcentaje de usuarios permitidos.

En definitiva, usar un sistema de seguridad basado en huella dactilar depende del nivel de seguridad y de la calidad de las huellas que se pueden utilizar.

En sistemas en los que no se puede asegurar que la calidad de la huella pueda tener al menos un NFIQ inferior a 3, no es aconsejable usar un sistema biométrico. Para NFIQ tan altos tan sólo pueden llegar a ser usados cuando se usa un umbral bajo, lo cuál hace que el número de impostores se dispare llegando al punto en que el porcentaje de usuarios no deseados es tan elevado que el más simple candado haría un mejor trabajo. Otra posibilidad es utilizar valor de umbral alto, lo que hace que la mayor parte de usuarios genuinos serán

rechazados lo que podría crear un cuello de botella en el sistema de seguridad. Por encima de todo esto, la diferencia a este nivel de calidad entre umbral bajo y alto es muy pequeño, lo que hace que sea muy difícil regular este sistema.

Si se puede tener seguridad en que la calidad de las huellas va a estar por debajo de un NFIQ de 3 entonces estos sistemas de huella dactilar pueden ser de gran utilidad. Si se quiere un valor de exigencia medio con un rechazo del 90% de usuarios impostores se pueden utilizar todas estas huellas de forma indiscriminada. Pero, incluso si se quiere un sistema cuyo rechazo esté entre el 95% e incluso superior se podría seguir utilizando estos sistemas con calidades de huella más altas (NFIQ 2 ó 1).

7.2 Comparativa entre los sensores

Puesto que los resultados de cada sensor han sido más dispares de lo esperado se puede hacer una comparativa entre ellos para ver qué sensor tiene mejores prestaciones en cada momento.

Un dato interesante es ver que, a pesar de variar en rendimiento, la diferencia entre los sensores es muy similar independientemente de la calidad utilizada. Podría pensarse que un sensor podría tener mejores prestaciones que otro en calidades medias mientras que otro podría funcionar mejor en calidades altas pero no ha sido así. Según los datos obtenidos el sensor NXT tiene unas prestaciones bastante más altas que el sensor FPC y UPK. Las diferencias en rendimientos se disminuyen a calidades medias pero igualmente hay un claro vencedor.

Puesto que el estudio se centra en la calidad de las huellas y la influencia que estas tienen en los sensores también hay que observar si cada sensor se ve influenciado de la misma forma con cada nivel de calidad. Aquí vemos similitud entre los sensores UPK y FPC, ambos sufren muy poco en la diferencia entre la Calidad Media y la Calidad Alta, ambas calidades tienen prestaciones muy similares en especial en el sensor FPC en el cuál prácticamente no existe diferencia ninguna. El sensor NXT sí que tiene una diferencia más

notable entre estas calidades lo cual hace pensar que el sensor NXT es más sensible a la calidad de sus muestras. Para los tres sensores sí es claro que los valores de NFIQ 4 y 5 son devastadores, habiendo una gran diferencia entre la Calidad Baja y el resto de calidades estudiadas.

En cuanto a por qué el sensor NXT muestra resultados mejor que los sensores UPK y FPC habría que plantearse la tecnología usada en esto sensor. El sensor NXT es termosensible mientras que los sensores UPK y FPC son de tipo capacitivo. Es posible que o bien los valores NFIQ no se adaptan bien a todo tipo de tecnología, o bien los sensores térmicos hayan demostrado superar a los capacitivos. Si tomamos el primer caso, explicaría el comportamiento del sensor NXT en las calidades media y alta, donde los valores llegan a confundirse.

7.3 Líneas Futuras

Aunque este estudio puede aportar bastante luz a la influencia de la calidad de las muestras en los sistemas biométricos aún se puede ampliar enormemente para tener una mejor visión sobre el mundo de la biometría. Algunos de estos estudios podrían ser:

- Influencias sobre la calidad de las huellas: Un comentario realizado durante las conclusiones hace referencia a que si se puede asegurar un nivel de calidad inferior a un NFIQ de 4 los sistemas mejoran enormemente. Un estudio a realizar sería ver qué factores influyen sobre la calidad de la imagen. Si el estudio permitiese ver en qué ocasiones se puede asegurar una calidad de huella óptimo esto permitiría hacer una mejor selección de cuando usar un sistema biométrico.

- Algoritmo NFIQ: Lo visto en este trabajo muestra que la diferencia entre las calidades Media y Alta resulta muy baja con respecto a la Calidad Baja. Un estudio que podría hacer más fiable el valor NFIQ sería comprobar como determina qué valor asignar a cada imagen y modificarlo para que la diferencia

entre valores estuviese distribuido de mejor manera, de esta forma hay mayor cabida a distintos niveles de seguridad según la calidad de la imagen.

---

8. BIBLIOGRAFÍA

[1] http://www.xatakaciencia.com/biologia/como-se-sabe-que-todas-las-huellas-dactilares-son-diferentes-en-cada-persona Referencia al libro Galton, Francis (1892). Finger Prints. Londres.

[2] DUNSTONE, T. y YAGER, N. (2009). *Biometric System and Data Analysis*, Ciudad de la empresa editora del libro: Springer.

[3] http://www.nist.gov/public_affairs/general_information.cfm (Septiembre 2015).

[4] http://www.nist.gov/itl/iad/ig/nbis.cfm (Septiembre 2015)

[5]ISO/IEC FDIS 19795-1. "Information technology – Biometric performance testing and reporting – Part 1: Principles and framework". Normativa de estudios biométricos.

[6] http://file.scirp.org/Html/3-9701272_3692.htm (Septiembre 2015)

[7] http://gim.unmc.edu/dxtests/roc3.htm (Septiembre 2015)

[8] Busch, Christoph "From Liveness Detection to Presentation Attack Detection" (2014) Oslo.

[9] http://www.nist.gov/itl/iad/ig/bws.cfm. (Mayo 2015)

[10] http://stackoverflow.com/questions/6355135/c-sharp-converting-32bpp-image-to-8bpp (Mayo 2015)

[11]FERNÁNDEZ SAAVEDRA, María Belén. "Performance Testing Evaluation Report of Results." (2015) Madrid: GUTI. Documento de estudio de rendimiento de sensores biométricos.

[12] FERNÁNDEZ SAAVEDRA, María Belén. "Documentación del curso sobre la biometría y su evaluación" (2014). Madrid: UC3M. Documento no publicado.

9. ANEXOS

9.1 Planificación

El estudio fue planteado por primera vez en Enero de 2015 lo cual ha permitido que el proceso de estudio teórico se haya realizado durante un largo periodo de tiempo.

Para poder entender el estudio que se iba realizar hubo una primera temporada en la que se estudió los documentos existentes sobre biometría. Una vez entendido como funciona la biometría se comenzó a realizar la aplicación en C# que crearía las listas FAR y GAR. Al conseguir unas listas con los datos necesarios estas se transformaron en las gráficas de rendimiento utilizando el programa MatLab. Finalmente, se pudo hacer un análisis sobre estas gráficas (Tabla 1).

Fecha	Descripción	Tiempo
02/02/2015	Curso de biometría impartido por la doctora María Belén Fernández Saavedra	1 día
10/06/2015	Lectura de la documentación sobre sistemas biométricos y estudios de rendimiento de sistemas por huella dactilar	2 semanas
01/07/2015	Programación en C# de creación de listas FAR y GAR	4 semanas
29/07/2015	Primera prueba con la base completa de usuarios	3 días
03/08/2015	Modificaciones finales del programa en C#	1 semana
10/08/2015	Ejecución del programa de creación de lista con los 50 usuarios	2 días
12/08/2015	Representación en MatLab de los datos creados	1 semana
19/08/2015	Análisis de los resultados	1 semana

Tabla 1: Planificación

9.2 Presupuesto

El presupuesto se divide en dos capítulos: Materiales y salarios (Tabla 2). El estudio carece de un gran número de materiales para ser llevado, la mayor parte del presupuesto se va en el salario del ingeniero que ha realizado el estudio.

La base de datos proporcionada por el GUTI tiene un precio base por usuario. Estos usuarios se componen de imágenes de Reclutamiento e imágenes de verificación, ambos tipos de imagen por cada sensor utilizado. El precio del ordenador Intel se ha tenido en cuenta que se puede seguir utilizando para estudios futuros, es por ello que el precio aparente en el presupuesto tan sólo incluye el precio de uso del ordenador durante estos meses de trabajo.

En lo que concierne al salario del ingeniero se ha tenido en cuenta que no se ha trabajado a jornada completa y que el tiempo en el que se ha realizado el estudio se ha visto alargado en cierta medida debido a que se iban adquiriendo conocimientos según se avanzaba, en vez de contratar a un ingeniero experto en la materia.

CÓDIGO	Medida	RESUMEN	CANTIDAD	PRECIO	IMPORTE
1		Material			
1.1	USUARIO	Base de datos: Base de datos de huella dactilar del GUTI. Esta base de datos proporciona varias imágenes de Reclutamiento y Verificación por cada usuario	50	13,50 €	675,00 €
1.2	UNIDAD	Ordenador: Ordenador Intel i5, 4GB RAM, 500 GB disco duro, Windows 8	1	150,00 €	150,00 €
		TOTAL: Apartado Material			825,00 €
2		Salario			
2.1	MES	Ingeniero: Graduado en ingeniería de electrónica Industrial y automática, sin experiencia previa y con conocimientos en programación C#	3	600,00 €	1.800,00 €
		TOTAL: Apartado Salario			1.800,00 €
		TOTAL PROYECTO sin IVA			2.625,00 €
				IVA	551,25 €
		TOTAL PROYECTO			3.176,25 €

Tabla 2: Presupuesto